NAVAL POSTGRADUATE SCHOOL

MONTEREY, CALIFORNIA

THESIS

AUTONOMOUS VISUAL TRACKING OF STATIONARY TARGETS USING SMALL UNMANNED AERIAL VEHICLES

by

Robert A. Prince

June 2004

Thesis Advisor:	Isaac I. Kaminer
Second Reader:	Ick Ho Whang

Approved for public release; distribution is unlimited

THIS PAGE INTENTIONALLY LEFT BLANK

REPORT DOCUMENTATION PAGE			Form Approved OMB No. 0704-0188
Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instruction, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188) Washington DC 20503.			
1. AGENCY USE ONLY ***(Leave blank)***	**2. REPORT DATE** June 2004	**3. REPORT TYPE AND DATES COVERED** Master's Thesis	
4. TITLE AND SUBTITLE: Autonomous Visual Tracking of Stationary Targets Using Small Unmanned Aerial Vehicles			**5. FUNDING NUMBERS**
6. AUTHOR(S) Robert Adam Prince			
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Naval Postgraduate School Monterey, CA 93943-5000			**8. PERFORMING ORGANIZATION REPORT NUMBER**
9. SPONSORING /MONITORING AGENCY NAME(S) AND ADDRESS(ES) N/A			**10. SPONSORING/MONITORING AGENCY REPORT NUMBER**
11. SUPPLEMENTARY NOTES The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.			
12a. DISTRIBUTION / AVAILABILITY STATEMENT Approved for public release; distribution is unlimited			**12b. DISTRIBUTION CODE**

13. ABSTRACT (maximum 200 words)

A control system was developed for autonomous visual tracking of a stationary target using a small unmanned aerial vehicle. The kinematic equations of this problem were developed, and the insight obtained from examination was applied in developing controllers for the system. This control system controlled the orientation of the camera to keep it constantly pointing at the target, and also controlled the trajectory of the aircraft in flight around the target. The initial control law that was developed drives the aircraft trajectory to a constant radius around the target. The range to the target is not directly measurable, so it was estimated using steady state Kalman filters. Once a range estimate is obtained, it is used to control the range to the target, and the aircraft trajectory is driven toward a circle with a specified radius. Initial tests of the control system with Simulink simulations have shown good performance of the control system. Further testing with hardware will be conducted, and flight tests are scheduled to be conducted in the near future. Conclusions are drawn and recommendations for further study are presented.

14. SUBJECT TERMS Unmanned Aerial Vehicle, UAV, Autonomous Guidance, Target Tracking, Control, Visual Target Tracking, Range Estimation, Triangulation, Kalman Filter, Piccolo, xPC Target			**15. NUMBER OF PAGES** 91
			16. PRICE CODE
17. SECURITY CLASSIFICATION OF REPORT Unclassified	**18. SECURITY CLASSIFICATION OF THIS PAGE** Unclassified	**19. SECURITY CLASSIFICATION OF ABSTRACT** Unclassified	**20. LIMITATION OF ABSTRACT** UL

NSN 7540-01-280-5500 Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std. 239-18

THIS PAGE INTENTIONALLY LEFT BLANK

Approved for public release; distribution is unlimited

AUTONOMOUS VISUAL TRACKING OF STATIONARY TARGETS USING SMALL UNMANNED AERIAL VEHICLES

Robert A. Prince
Ensign, United States Naval Reserve
B.S.E, Duke University, 2003

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN MECHANICAL ENGINEERING

from the

NAVAL POSTGRADUATE SCHOOL
June 2004

Author: Robert A. Prince

Approved by: Isaac I. Kaminer
Thesis Advisor

Ick Ho Whang
Second Reader

Anthony J. Healey
Chairman, Department of Mechanical and
Astronautical Engineering

THIS PAGE INTENTIONALLY LEFT BLANK

ABSTRACT

A control system was developed for autonomous visual tracking of a stationary target using a small unmanned aerial vehicle. The kinematic equations of this problem were developed, and the insight obtained from examination was applied in developing controllers for the system. This control system controlled the orientation of the camera to keep it constantly pointing at the target, and also controlled the trajectory of the aircraft in flight around the target. The initial control law that was developed drives the aircraft trajectory to a constant radius around the target. The range to the target is not directly measurable, so it was estimated using steady state Kalman filters. Once a range estimate is obtained, it is used to control the range to the target, and the aircraft trajectory is driven toward a circle with a specified radius. Initial tests of the control system with Simulink simulations have shown good performance of the control system. Further testing with hardware will be conducted, and flight tests are scheduled to be conducted in the near future. Conclusions are drawn and recommendations for further study are presented.

THIS PAGE INTENTIONALLY LEFT BLANK

TABLE OF CONTENTS

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF FIGURES

LIST OF TABLES

THIS PAGE INTENTIONALLY LEFT BLANK

ACKNOWLEDGMENTS

I would like to thank Professor Isaac Kaminer for allowing me to work on this interesting project and for all of his assistance along the way. Also, thanks to Dr. Ick Ho Whang for his countless hours of explanations and his constant willingness to help me understand various aspects of this project. I would also like to thank Dr. Vladimir Dobrokhodov for all of his help in implementing the ideas that were developed into working software. Also, thanks to Professor Kevin Jones for all of the work he has done to constructing the aircraft and the rest of the physical systems that will be used in actual flight tests.

On a more personal note, I would like to thank all of the other Ensigns who have made my time in Monterey so enjoyable. Finally, I would like to thank my parents who have supported me in everything I have done and have helped me get to where I am today.

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF ACRONYMS, SYMBOLS, AND SUPERSCRIPTS

ACRONYMS

UAV	Unmanned Aerial Vehicle
LOS	Line of Sight
PTU	Pan-Tilt Unit
DOF	Degree of Freedom
GPS	Global Positioning System
IMU	Inertial Measurement Unit
PI	Proportional plus Integral

SYMBOLS

ϕ	roll angle
θ	pitch angle
ψ	yaw angle
p	rotation rate around the x-axis
q	rotation rate around the y-axis
r	rotation rate around the z-axis
λ	angle between the x-axis of the inertial frame and the line of sight vector to the target
ε	angle between the camera heading and the target line of sight vector
η	angle from the line perpendicular to the line of sight vector to the velocity vector

SUPERSCRIPTS AND SUBSCRIPTS

I	Inertial Reference Frame
B	Body Reference Frame
G	Gimbal Reference Frame
C	Camera Reference Frame
P	Image Plane Reference Frame

ANGLE REPRESENTATION

Angles are represented as ${}^{B}A_{C}$

Where A Type of angle (ϕ, θ, ψ)

B Frame of Reference

C Frame being referenced

Therefore ${}^{I}\psi_{B}$ represents the yaw angle of the aircraft body in reference to the inertial frame

THIS PAGE INTENTIONALLY LEFT BLANK

I. INTRODUCTION

A. BACKGROUND

The goal of the project was to support of the Surveillance and Target Acquisition Network (STAN) Project being conducted at the Naval Postgraduate School, which is focused on developing increased network-centric warfare capabilities. This is done by creating a network of multiple platforms that are able to communicate necessary information. One of the platforms included in this network is a small Tern UAV, most often being used for target reconnaissance and surveillance. In many cases, the mission of a UAV is to locate a target and continuously track it. Information about the target can then be shared amongst the communications network allowing for tactical decisions to be made and providing necessary information to other equipment.

The current process for visually tracking a target with a pan-tilt camera mounted on a UAV requires two operators. One operator controls the UAV while the second controls the orientation of the camera. These two people must work together and continuously communicate in order to keep the camera aimed at the target. With modern computers and commercially available tracking software, there is no reason that this process cannot be automated. Obviously UAV's cannot operate completely autonomously because there must be some human element in the loop to make decisions and take over if unexpected problems are encountered, but much of the control process can be done with computers. If computers can do the majority of the work in controlling the system, the operator can focus more on the mission at hand. Also, if much of the process is automatic, one operator could feasibly operate multiple vehicles at the same time.

B. PROBLEM FORMULATION

The purpose of this project was to develop and demonstrate the ability to autonomously track a target from a small UAV equipped with a pan-tilt camera. This required the development of the actual physical system, and more importantly, the development of the control laws that govern the system. Both of these problems will be discussed in this thesis, however the main focus was on the development and testing of the control system.

The first step was to acquire all of the necessary equipment and assemble it so that everything was compatible. This required the selection of the proper aircraft platform for the equipment to be installed on, a pan-tilt unit, a camera, aircraft autopilot, image tracking software, and multiple computers to run all of the software. The equipment, as well as the method for combining everything into a working system, will be discussed in detail in the following chapters.

As stated earlier, the main focus of this project was the development of the control system for autonomous tracking. In order to develop the control system, it was first necessary to examine the kinematics of the problem. Simplified models were created that represented the relationship between the aircraft and the target and also between the pan-tilt camera and the target. These simplified models were then used in the development of a control law that does three things. First, the UAV would be forced to circle the target while keeping the camera pointed directly at it. While circling, an estimate of the range to the target is created, and this estimate is then used to allow the user to increase or decrease the range. The development of the control theories will be discussed in later chapters, as will the theory and method for range estimation.

After the control system was developed in simplified models, it was necessary to test its applicability to the real situation. Models of all the components were created in Simulink, including a six degree of freedom model for the aircraft and autopilot and models of the pan-tilt unit, camera, and image processing software. All of these models were connected together to create a realistic simulation of actual problem. To further test the control laws in a more realistic environment, hardware in the loop (HIL) simulations were conducted with the image processing software being incorporated in the system. Finally, flight tests are scheduled to be conducted in the near future to verify that everything works in a real-life situation.

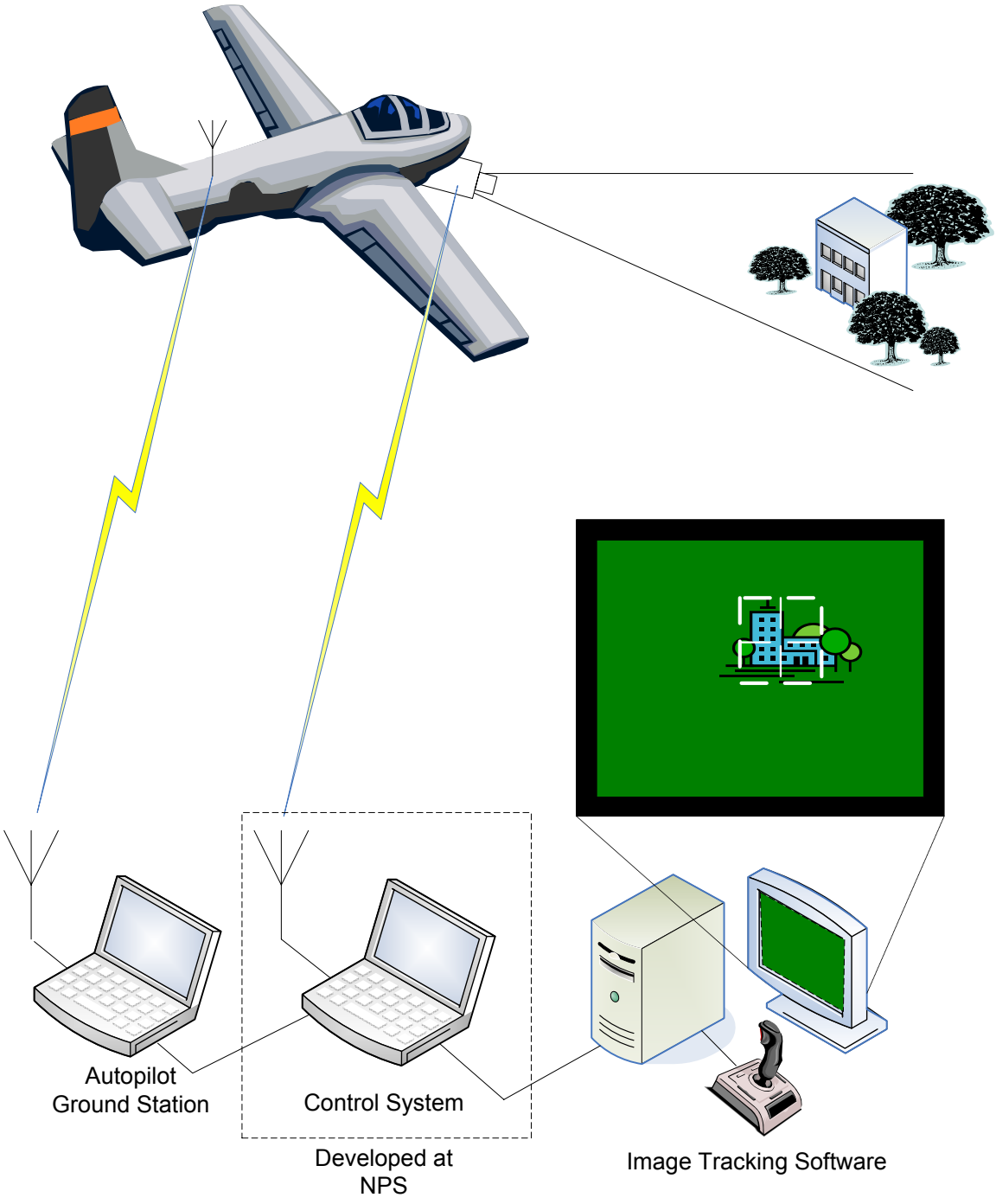

Figure 1. Overview of Visual Target Tracking Problem

THIS PAGE INTENTIONALLY LEFT BLANK

II. COORDINATE SYSTEMS AND TRANSFORMATIONS

A. COORDINATE SYSTEMS

In complex dynamic systems it is common to use multiple coordinate systems because it is often easier to define the motion of an object in one frame rather than another. As long as the relationships between the coordinate frames are known, the positions, velocities, and accelerations can be transformed between coordinate frames to describe the true motion in any frame. The following coordinate systems were used in the development of this problem.

1. Inertial Coordinate Frame (I-Frame)

In this thesis, a flat earth assumption has been made and the Local Tangent Plane is used to represent the inertial coordinate system. Although the use of a fixed frame is a simplifying assumption, it is entirely valid for this problem because everything is limited to one local area. This is a north-east-down (NED) coordinate system in which the x-axis corresponds to north, the y-axis corresponds to east, and the z-axis is directed downward toward the center of the earth.

2. UAV Body Coordinate Frame (B-Frame)

The UAV body coordinate frame is a right-hand orthogonal system with the origin at the aircraft center of gravity. The x-axis is aligned with the longitudinal axis of the aircraft, the y-axis is aligned with the right wing, and the z-axis is in the downward direction.

3. Gimbal Platform Coordinate Frame (G-Frame)

The gimbal coordinate frame is a right-hand orthogonal coordinate system with the origin being the location of the camera mount. The x-axis of the gimbal frame is aligned with the gimbal platform, the y-axis points outward toward the right hand side, and the z-axis points downward from the gimbal platform.

4. Camera Coordinate Frame (C-Frame)

The camera coordinate frame is a right-hand orthogonal coordinate system with the origin located at the focal point of the camera. The x-axis is located along the longitudinal axis of the camera, the y-axis points outwards toward the right hand side, and the z-axis points downward from the origin.

5. Image Plane Coordinate Frame (P-Frame)

The image plane reference frame is the coordinate system used to describe the location of the target in the image plane. This is a 2-dimensional coordinate system with the u-axis aligned with the y-axis of the camera frame, and the v-axis aligned with the negative z-axis of the camera frame.

B. EULER ANGLES

The most common method for defining angular orientation of one coordinate system in respect to another coordinate system is through the use of three Euler angles; ϕ, θ, and ψ. When used to describe the orientation of the aircraft body in relation to the inertial frame, these Euler angles are known as roll, pitch, and yaw respectively. Euler angles are effective in uniquely defining the relative orientation of two coordinate frames as long as θ does not approach 90°, at which point the orientation is not uniquely defined. This did not cause any problems in the situations encountered in this project, but if problems were to be encountered the slightly more complicated quaternion representation could be used.

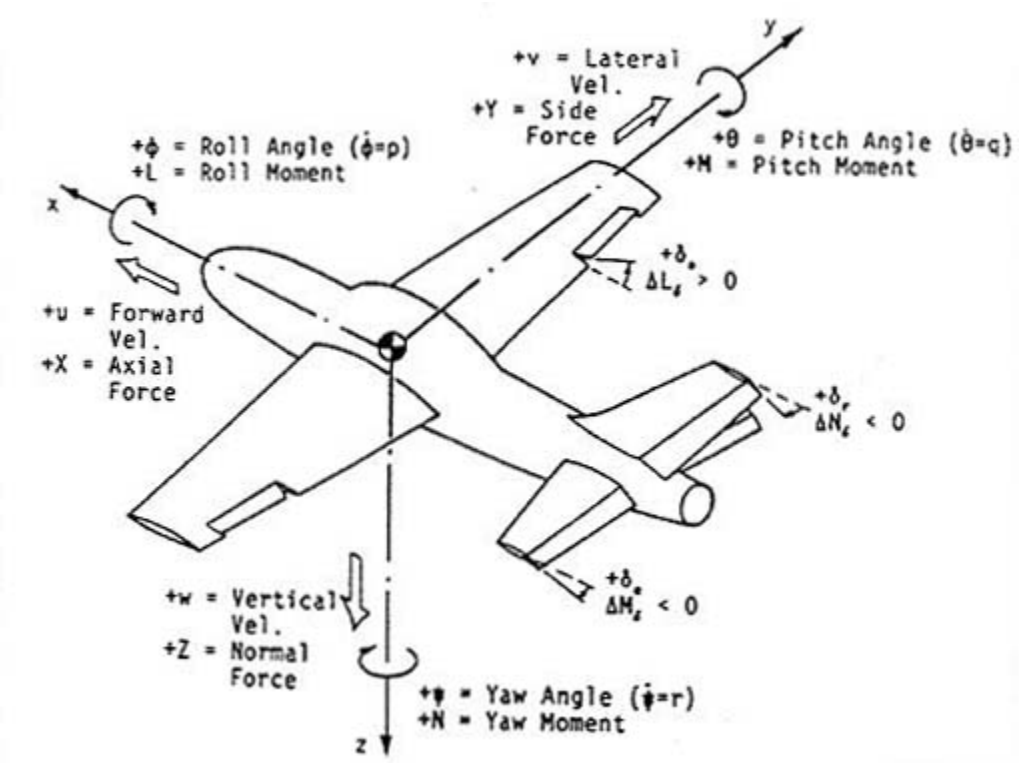

Figure 2. Illustration of Euler Angles (From: Ref. 4)

C. RELATIONSHIPS BETWEEN COORDINATE FRAMES

In order to appropriately define the orientation of a rigid body in space, it is necessary to define both position and orientation. Therefore at any instant in time, the position can be defined by a [3 x 1] vector and the orientation can be defined by a [3 x 3] rotation matrix.

1. Rotation Matrices

The rotation matrices for rotation about each individual axis are given below. The angle of rotation is the Euler angle that corresponds to each individual axis.

$$R_x(\phi) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\phi & \sin\phi \\ 0 & -\sin\phi & \cos\phi \end{bmatrix} \tag{1}$$

$$R_y(\theta) = \begin{bmatrix} \cos\theta & 0 & -\sin\theta \\ 0 & 1 & 0 \\ \sin\theta & 0 & \cos\theta \end{bmatrix} \tag{2}$$

$$R_z(\psi) = \begin{bmatrix} \cos\psi & \sin\psi & 0 \\ -\sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{3}$$

The actual rotation matrix that is used in defining the orientation of the body is usually a combination of all three of these, however because of some special situations they will be rigorously defined.

2. Coordinate Transformations

The coordinate transformation from the inertial frame to the camera frame can be obtained by multiplying each of the individual coordinate transformations together in the correct order as shown here.

$$^{C}_{I}C = {}^{C}_{G}C\,{}^{G}_{B}C\,{}^{B}_{I}C \tag{4}$$

a. Inertial Frame to Body Frame

The coordinate transformation from the inertial frame to the body frame is simply the product of the three individual rotation matrices.

$$ {}^{B}_{I}C = R_x\left({}^{I}\phi_B\right) R_y\left({}^{I}\theta_B\right) R_z\left({}^{I}\psi_B\right) \tag{5} $$

b. Body Frame to Gimbal Platform Frame

The coordinate transformation from the body frame to the gimbal platform frame only involves rotation through two angles because it is a two-axis gimbal.

$$ {}^{G}_{B}C = R_y\left({}^{B}\theta_G\right) R_z\left({}^{B}\psi_G\right) \tag{6} $$

c. Gimbal Platform Frame to Camera Frame

The coordinate transformation between the gimbal platform frame and the camera frame allows for any misalignment between the camera and the gimbal platform. It is the product of all three rotation matrices, however it is likely that some or all of the rotation angles will be zero because the axes will directly correspond.

$$ {}^{C}_{G}C = R_x\left({}^{G}\phi_C\right) R_y\left({}^{G}\theta_C\right) R_z\left({}^{G}\psi_C\right) \tag{7} $$

d. Camera Frame to Image Plane Frame

The coordinate transformation between the camera frame and the image plane frame is not a rotation matrix like the others. Instead it is a position transformation that specifies the position of an object in the image plane frame given the position in the camera frame.

$$ \begin{bmatrix} {}^{P}u \\ {}^{P}v \end{bmatrix} = \frac{f}{{}^{C}x} \begin{bmatrix} {}^{C}y \\ -{}^{C}z \end{bmatrix} \tag{8} $$

Because the transformation from the camera frame to the image plane frame is a transformation between a three-dimensional coordinate system to a two-dimensional coordinate system, it is not possible to have a reverse transformation. Therefore, given a position in the camera frame, it is possible to determine the position in the image plane frame, but there are an infinite amount of possibilities if one attempts to locate a position from the image plane frame in the camera frame.

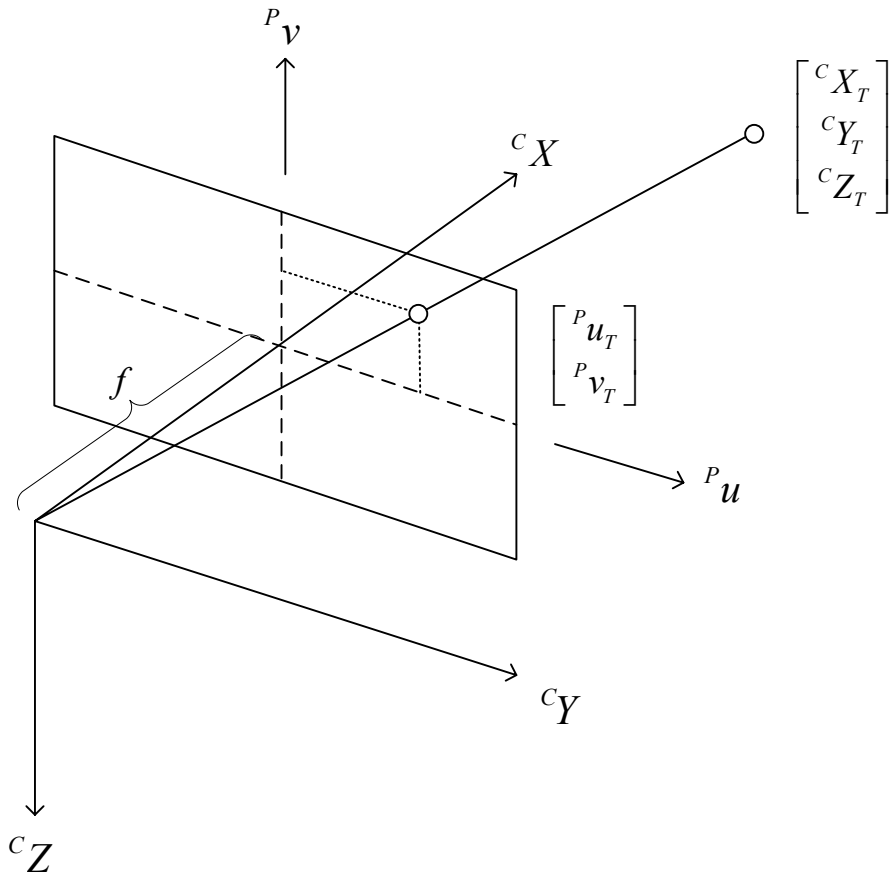

Figure 3. Image Plane Reference Frame

3. Angular Velocities

a. Angular Rate of Body Frame with Respect to Inertial Frame

Generally, p, q, r, are given by UAV angular motion dynamics model. For reference, their relationships with Euler angle rates are stated here.

$$ {}^{B}\omega_{BI} \triangleq \begin{bmatrix} p \\ q \\ r \end{bmatrix} = \begin{bmatrix} \dot{\phi} - \dot{\psi}\sin\theta \\ \dot{\theta}\cos\phi + \dot{\psi}\cos\theta\sin\phi \\ \dot{\psi}\cos\theta\cos\phi - \dot{\theta}\sin\phi \end{bmatrix} \tag{9} $$

These equations are not valid if pitch angles will approach + or – 90 degrees, but this will not be an issue for this problem so a quaternion representation will not have to be used.

b. Angular Rate of Gimbal Platform Frame with Respect to Inertial Frame

The angular rate of the gimbal platform frame with respect to the inertial frame is expressed in the equations below. The first describes the angular rate expressed in the gimbal platform frame while the second describes the same angular rate in the inertial frame.

$$ {}^{G}\omega_{GI} \triangleq \begin{bmatrix} p_G \\ q_G \\ r_G \end{bmatrix} = {}^{G}_{B}C \begin{bmatrix} p \\ q \\ r \end{bmatrix} + {}^{G}_{B}C \begin{bmatrix} 0 \\ 0 \\ \dot{\psi}_G \end{bmatrix} + R_y\left(\theta_G\right) \begin{bmatrix} 0 \\ \dot{\theta}_G \\ 0 \end{bmatrix} \tag{10} $$

$$ {}^{I}\omega_{GI} = {}^{I}_{G}C\,{}^{G}\omega_{GI} = {}^{I}_{G}C \begin{bmatrix} p_G \\ q_G \\ r_G \end{bmatrix} \tag{11} $$

c. Angular Rate of Camera Frame with Respect to Inertial Frame

Since there is no rotation between the camera frame and the gimbal platform frame, the angular rate of the camera frame is the same as the gimbal frame as long as there is perfect alignment. Therefore, the angular rate between the camera frame and the inertial frame is given by the two equations below. The first describes the relationship to the camera frame, and the second describes the angular rate in the inertial frame.

$$ {}^{C}\omega_{CI} = {}^{C}_{G}C\,{}^{G}\omega_{GI} = {}^{C}_{G}C \begin{bmatrix} p_G \\ q_G \\ r_G \end{bmatrix} \tag{12} $$

$$ {}^{I}\omega_{CI} = {}^{I}\omega_{GI} \tag{13} $$

III. SENSORS, EQUIPMENT, AND NETWORK PROTOCOL

A. OVERVIEW

The purpose of this project was to develop the control system to allow for autonomous tracking of a target with a small UAV. In order to test that the ideas developed in this project are valid, and in order to demonstrate their effectiveness, they had to be applied to a physical system. The aircraft was built to accommodate the required payload, which included the pan-tilt unit, camera, and piccolo autopilot. The ground station also had to be put together which included the piccolo ground station along with its host computer, a PC104 computer running xPC target software, and a computer running the PerceptiVU image tracking software. All of this equipment was networked together so that data could be shared between the computers.

B. PICCOLO AUTOPILOT

The Piccolo integrated avionics system from Cloud Cap Technologies is used as the interface to control the UAV in all modes of operation. By itself, it is designed to be used as an autopilot system with an inner control loop within the aircraft itself, and an outer control loop provided from the ground station. Once connected with the rest of the equipment used in this project it provides the necessary communication for implementing the guidance control law. All information regarding Piccolo is taken from Ref. 6.

Figure 4. Piccolo avionics mounted onboard aircraft

1. Control System Setup

The piccolo control system consists of four main parts: an avionics control system that is mounted onboard the UAV, a ground station, a computer for operator interface, and a pilot manual control interface. With these components alone, the user can program desired routes for the aircraft to fly via waypoint guidance. The manual control interface is to allow the user to take control and fly the aircraft, which is especially useful during takeoff and landing.

The control setup for the piccolo system has two separate control loops. The fast inner loop controls the aircraft dynamics and takes place within the aircraft itself. The slower outer loop is a wireless communication between the UAV and the ground station which dictates the path that the aircraft is expected to follow.

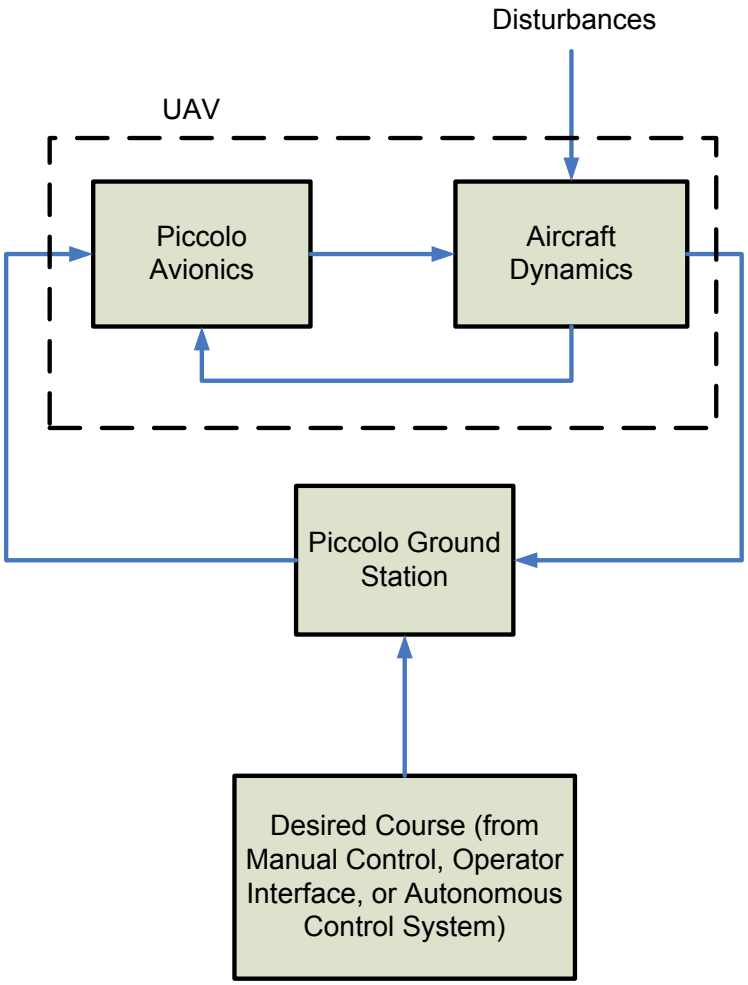

Figure 5. Inner and Outer Loops of Piccolo Autopilot

2. Avionics

The piccolo avionics system is the onboard component of the overall control system. It is the autopilot for the aircraft that is designed to receive the input of the commanded path from the control station and generate the required signals to move the control surfaces of the aircraft (ailerons, elevator, rudder, and throttle).

a. Processor

The CPU for the Piccolo avionics is the MPC555 microcontroller. This allows for an enormous array of interfaces and delivers 40 Mhz PowerPC operation. The processor controls everything within the avionics and performs sophisticated filtering of all the data in order to obtain accurate, reliable results.

b. Rate Gyros

Piccolo is equipped with three Tokin CG16D rate gyros. These gyros can be mounted at any attitude due to CPU's ability to run sophisticated filters on the collected data.

c. Accelerometers

Piccolo is also equipped with two two-axis ADXL202 accelerometers. These allow for acceleration measurements in all three directions, which are fed to the CPU and filtered along with measurements from the GPS and rate gyros.

d. GPS

Motorola M12 GPS provides basic groundspeed and position. The M12 is differential capable, and it is supplied with and DGPS corrections received over the datalink. These corrections are generated by the piccolo ground station which is also equipped with an M12 GPS

e. Pressure Sensors

Dual ported mpxv50045 4 kPa dynamic pressure sensor, an absolute ported mpx4115a barometric pressure sensor, and a board temperature sensor. Together these sensors provide the ability to measure true air speed and altitude.

3. Ground Station

The ground station is based upon the same hardware as the avionics system. It manages the communication with the aircraft, interfaces to the pilot in the loop control, and provides a command and control data stream to and from the operator interface PC.

4. Operator Interface

The operator interface is a software program that runs on a Windows PC and provides command and control interface for Piccolo operators. Most of the system features are accessible from this interface, including preflight setup, flight planning, and calibration of the sensors and surfaces of the aircraft.

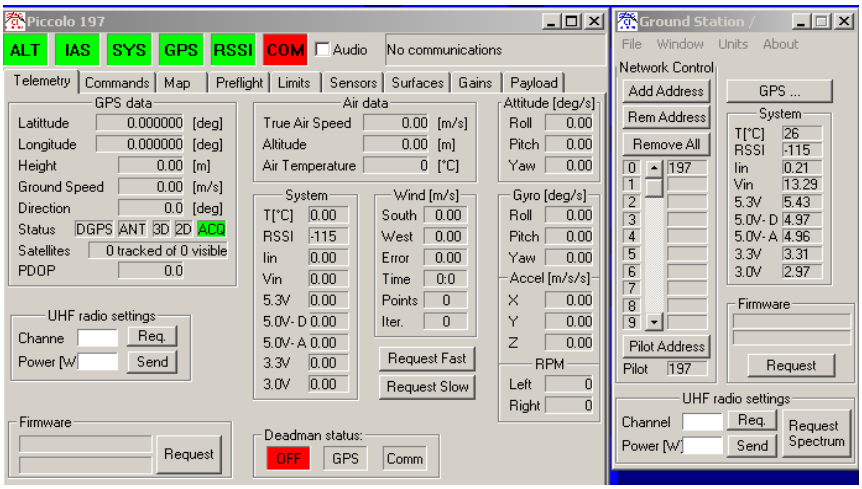

Figure 6. Screenshot of Piccolo Operator Interface

5. Pilot Manual Control

In order to fly the aircraft in a manual mode, a pilot in the loop mode can be activated. While in this mode, commands are input by the pilot with a Futaba controller that has been modified to communicate directly with the ground station. Control can be easily switched between the piccolo ground station and the pilot at the press of a button.

6. Piccolo Communications Protocol

The data communicated between devices within the piccolo system are encoded in a unique two-layer protocol. The inner layer consists of information about the size and type data followed by the data itself. The outer layer contains information about the avionics device that is sending or receiving the data, information about the size of the payload, and then the inner layer of information. Synchronization bytes are used

throughout to ensure that everything matches up and to determine if errors occurred during transmission. In order to send and receive data between the control system developed in this project and the piccolo autopilot, drivers had to be created to encode and decode the data into the necessary formats. All of this was done in previous work at NPS and is documented in Ref. 10.

C. AIRCRAFT AND PAYLOAD

1. Telemaster 40 R/C Airplane

The Telemaster 40 Radio Controlled airplane was selected as the aircraft to be used in this project. It was selected because it is a large, easy to fly aircraft and will be extremely stable. Furthermore, it was big enough to be able to carry the necessary payload of a pan-tilt unit, camera, and Piccolo autopilot. Because of the large size and inertia of the aircraft, it is not very susceptible to external disturbances.

Special care was taken during construction to isolate the engine from the airframe in order to limit vibrations. This is important because the success of the visual target tracking software is dependent on the quality of the video that is used. Unstable images with a lot of vibration would make it extremely difficult to track targets.

Figure 7. Telemaster 40 R/C Airplane

2. Directed Perception PTU-D46-70

The pan-tilt unit that was purchased for use in this project was the Directed Perception PTU-D46-70. This unit was deemed too heavy to actually be used on the

aircraft, however it is still being used for testing and hardware in the loop simulations until another pan-tilt unit can be created.

Figure 8. Directed Perception PTU-D46-70

3. Custom Built Pan-Tilt Unit

A custom built pan-tilt unit will be created for use in this system. This system will be light enough to mount onboard the aircraft for flight testing and will be powerful enough to control the position of the camera that will be mounted on it.

D. GROUND CONTROL HARDWARD AND SOFTWARE

1. PC-104/xPC Target

Developed by The MathWorks Inc., xPC target is a software program that allows models developed in Simulink and Real-Time Workshop to be downloaded and run in real time on a target computer. The target computer used here was a PC-104 stackable computer system. Essentially it allows for the generated Simulink models to be run in real time, and is used for hardware in the loop simulations as well as the actual implementation of the control system during flight testing.

2. PerceptiVU Image Tracking Software

The image tracking software used in this project was purchased from PerceptiVU, Inc. The software was installed on a PC computer equipped with a Matrox graphics card and extra DDRAM. A joystick was also incorporated into the system and allows the user to control the position of the pan-tilt unit and also select targets on the screen. The

software takes the analog video received from the aircraft and, when tracking a target, outputs the target offset position.

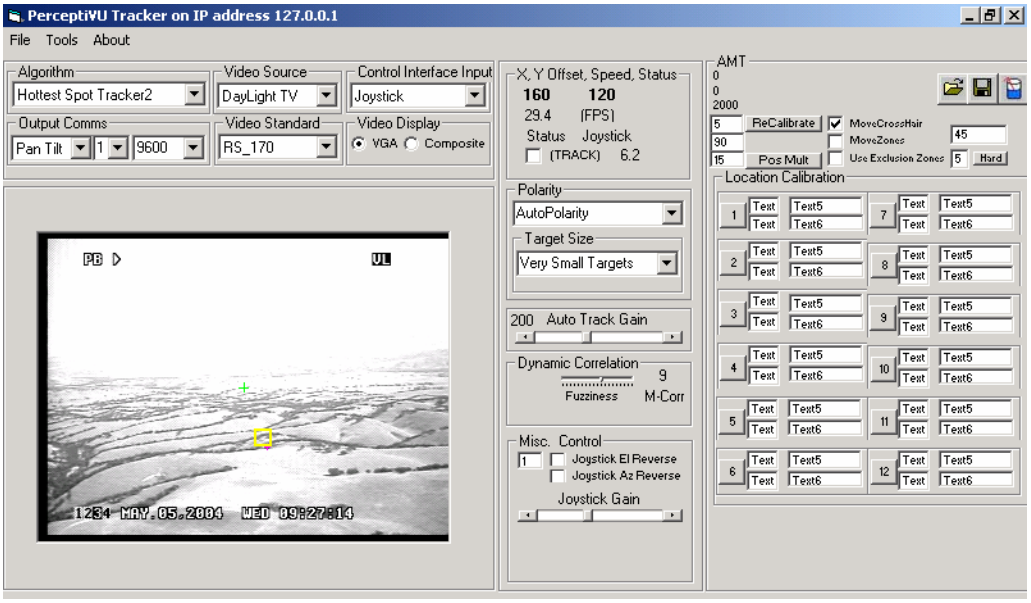

Figure 9. Screenshot of PerceptiVU Image Tracking Software

E. NETWORK PROTOCOL

1. UDP/IP

In order to communicate between all of the separate computers, User Datagram Protocol (UDP) was used. This was the best method for communicating in real-time among the computers to ensure that the most recent information was being sent and received at all times. Unlike TCP/IP, there is no verification that information was received. Information is still packed into data packets by the sender and unpacked by the receiver. The sender will send out information whenever it can, and the receiver will receive information whenever it is able to as well. Any information that is sent while the receiver is unavailable will simply be lost. In the problem developed here only real-time data is useful, so if the receiver is busy there is no reason to resend and slow down the network. This protocol is described as being connectionless and unreliable, however it is useful for the problem at hand because it will transfer the data in the fastest way possible.

THIS PAGE INTENTIONALLY LEFT BLANK

IV. DEVELOPMENT OF COMPLETE SYSTEM MODEL

A. OVERVIEW

In order to conduct realistic simulations to test the control laws developed in this thesis, it was necessary to create models of every aspect of the system. Where possible, care was taken to make the models as accurate as possible, but many simplifying assumptions were made to keep everything manageable. Appropriate system and measurement noises were added where they would be expected in the actual devices. Generalized models were first created, and then time constants and noise errors were supplied to make them specific to the devices that will actually be used.

B. MODEL OF PAN-TILT UNIT

The pan-tilt unit was modeled as a two-axis unit that could vary both the pitch angle and the yaw angle. Pitch and yaw rate commands are sent to the PTU and these commands are integrated within the motor to produce the pitch and yaw positions.

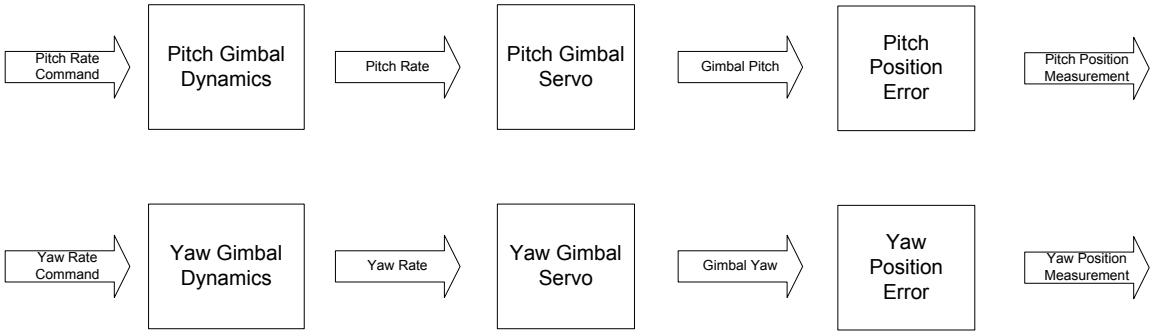

Figure 10. Block Diagram of Pan-Tilt Unit System Model

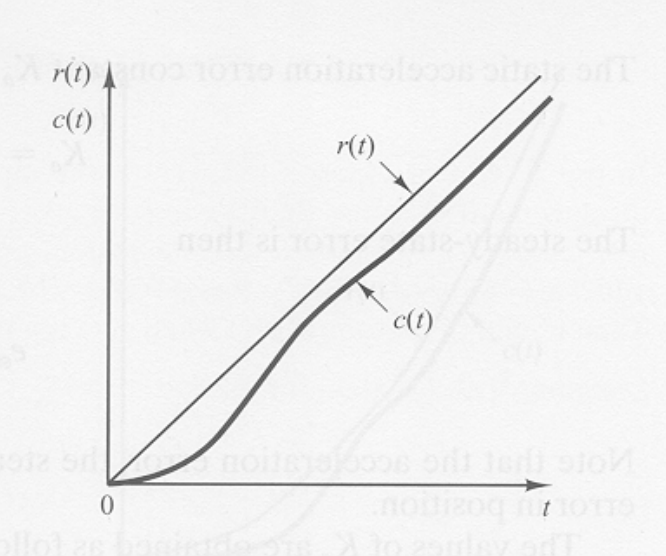

Figure 11. Illustration of Steady State Position Error for Constant Velocity Command (From: Ref. 3)

As is demonstrated in the block diagram, the input to the pan-tilt unit consists of both pitch and yaw rate commands. Because it is a physical system having inertia there is some inherent delay between the rate commands that are sent and the actual rate that is obtained.

The pan-tilt unit uses type 1 servos, which means there is one integrator, the motor, in the system. Therefore, if a constant velocity is commanded, that velocity will be obtained, but it will take a finite period of time to reach that velocity. This results in a steady state error for the position of the servo, but with fast acceleration the error can be kept to a minimum. In the problem at hand, it is expected that the desired position will change so frequently that steady state error won't be of any consequence.

The model was designed to have an error represented by a white noise process with a zero mean and a specified variance. In actual usage the pan-tilt unit may be more likely to slip in certain directions than others (i.e. downwards due to the weight of the camera) and this would have to be looked at and modeled more carefully. For now, the simple case of random error is used.

The transfer functions between the rate input and the position output of the pan-tilt unit servos are given as follows

$$\frac{Position(s)}{Velocity(s)} = \left(\frac{1}{.01s+1}\right)\left(\frac{1}{s}\right) = \frac{1}{.01s^2+s} \tag{14}$$

C. MODEL OF UAV AND AUTOPILOT

A six degree of freedom model of Silverfox UAV was used to model the UAV. Inner and outer loop autopilots were already designed, consisting of a simple altitude hold and turn coordination. The rigorous details are omitted here, but the development of this model can found in Ref. 10.

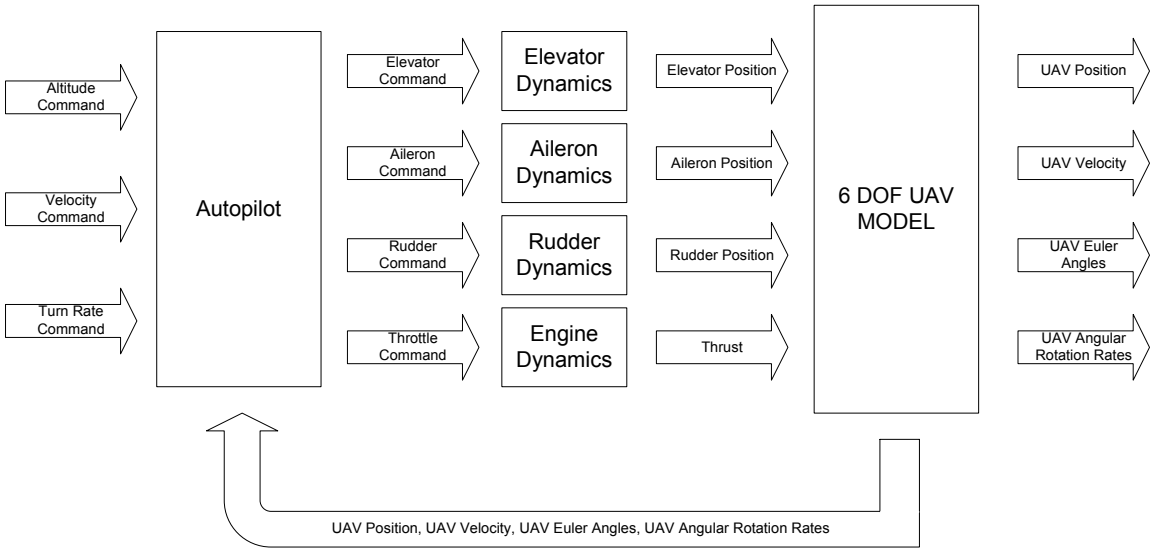

Figure 12. Block Diagram of Six Degree of Freedom Model and Autopilot

D. MODEL OF CAMERA

The camera was modeled simply as the representation of the target in the image reference plane. Using the known position and orientation of the UAV and assigning a position to the target, it is possible to determine

$$\begin{bmatrix} {}^{P}u \\ {}^{P}v \end{bmatrix} = \frac{f}{{}^{C}x}\begin{bmatrix} {}^{C}y \\ -{}^{C}z \end{bmatrix} \tag{15}$$

In order to use the above equation, it was first necessary to find the position of the target in the camera frame. The following two equations were used to determine the position of the target in the camera frame.

$${}^{C}\rho_{Tgt} = {}^{C}_{I}C\left({}^{I}_{B}C\,{}^{B}\rho_{Camera} + {}^{I}\rho_{UAV} - {}^{I}\rho_{Tgt}\right) \tag{16}$$

$$ {}^{C}_{I}C = {}^{I}_{C}C^{T} = \left({}^{B}_{I}C\, {}^{G}_{B}C\, {}^{C}_{G}C \right)^{T} \tag{17} $$

Where ${}^{B}\rho_{Camera}$ is the vector representing the position of the camera in the aircraft body frame, ${}^{I}\rho_{UAV}$ is the vector representing the position of the aircraft in the inertial frame, and ${}^{I}\rho_{Tgt}$ is the vector representing the position of the target in the inertial frame.

With the equations developed here, it was possible to obtain the position of the target in the image plane. In an ideal situation, this would be the actual position of the target within the camera image plane, but in real cameras there are often some distortions that would prevent this from being the case.

E. MODEL OF IMAGE PROCESSOR

The PerceptiVU image tracking software that was purchased is designed to autonomously track a target designated by the user. Therefore, it takes the target position in the image plane and creates rate commands that can be sent to the servos of the pan-tilt unit in order to drive the target position towards the center of the image plane. In order to do this, it measures the number of pixels that the image is offset from the center of the screen and uses these as values for u and v. These values are amplified by the pan-tilt control gain that is specified by the user, and are then output as the pan-tilt unit rate commands. For application to this problem it is more desirable to have u and v directly obtained from the image tracking software because this allows for better determination of the line of sight vector to the target. Modification of the software has to be done in order to get these outputs, so approximate methods for determination of the line of sight vector were also examined.

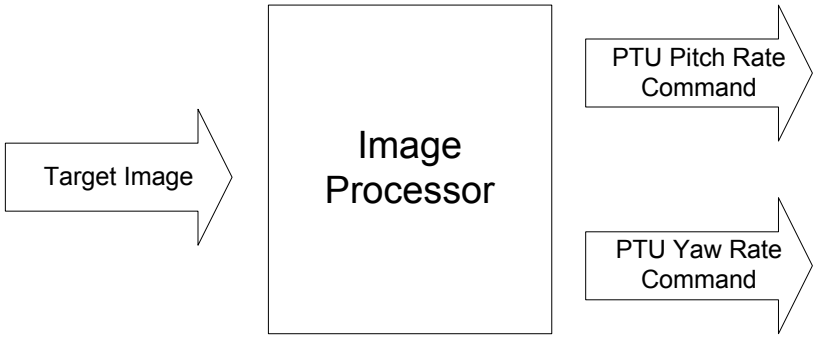

Figure 13. Image Processor Model

V. CONTROL SYSTEM DEVELOPMENT

A. OVERVIEW

The main focus of this project was the development of control laws to make autonomous visual tracking of stationary targets possible. The first step of this process was to analyze the underlying kinematics of the problem and gain insight on how control could be applied. It was also necessary to determine the control strategy for this problem. Finally, once a control system was developed, it was necessary to test it on a simplified model to verify its effectiveness.

It was decided that the control system should:

- Control the camera position to keep the target at the center of the image
- Control the aircraft to keep the target at the center of the image

A control system was then developed to accomplish these goals. First of all the control loop for the pan-tilt unit was developed that would send rate commands to move the camera position in order to keep the object in the center of the camera image. The next task was to develop control laws to govern the aircraft motion. In order to keep the target in the center of the camera image and allow the camera to easily track the target, a circular trajectory was chosen. With a circular trajectory of constant radius around the target the position of the camera can remain relatively stationary so the camera should not be required to move very much except to correct for disturbances.

For tactical reasons it was deemed necessary to be able to control the range between the target and the aircraft. Since the range to target is not directly measured by any of the sensors onboard the aircraft, it needs to be estimated. The methods for range estimation will be discussed in Chapter 6. Once a range estimate has been obtained, it can then be used by the control system to control the range.

The remainder of this chapter will discuss the formulation of the problem to determine what controllers were necessary, the development of the control laws, testing of the control laws with a simplified model, and the implementation of these control laws with the complete system.

B. PROBLEM FORMULATION

In development of the guidance control laws for both circular guidance and range control, it was necessary at first to create a simplified representation of the kinematics involved in the problem. This simplified model is presented below, with details on the assumptions and derivations presented in later sections.

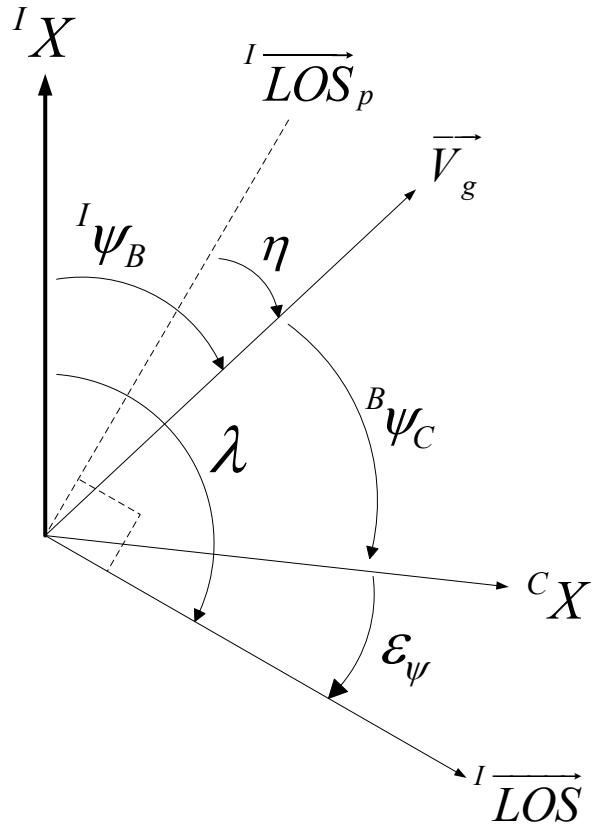

Figure 14. Relationships between angles in simple 2-dimensional kinematics model

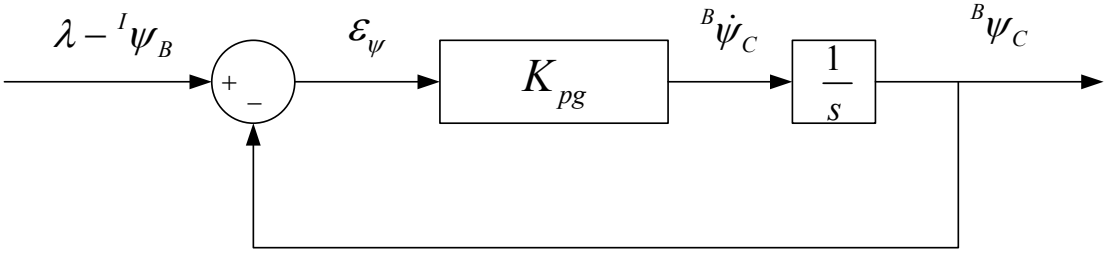

Figure 15. Servo Structure of Pan-Tilt Unit

1. Definition of Vectors

${}^{I}\overrightarrow{LOS}$ Ground line of sight vector to the target

$$ {}^{I}\overrightarrow{LOS} = {}^{I}_{C}C\,{}^{C}\overrightarrow{LOS} \tag{18} $$

${}^{I}\overrightarrow{LOS}_{p}$ Vector perpendicular to ground LOS vector

$$ {}^{I}\overrightarrow{LOS}_{p} = {}^{I}\overrightarrow{LOS} \pm \frac{\pi}{2} \tag{19} $$

$\overrightarrow{V_g}$ Ground velocity of aircraft

Directly measured from GPS

2. Definition of Angles

${}^{I}\psi_{B}$ Aircraft velocity heading

Directly measured from IMU

${}^{B}\psi_{C}$ Camera heading with reference to aircraft velocity

Directly measured from PTU position

ε_{ψ} Error angle of line of sight vector from camera heading

Directly measured from image tracking software

λ Angle of LOS vector in inertial frame

$$ \lambda = \varepsilon_{\psi} + {}^{I}\psi_{B} + {}^{B}\psi_{C} \tag{20} $$

η Angle between the perpendicular of the LOS vector and the velocity vector

$$ \eta = \sin^{-1}\frac{\left\| {}^{I}\overrightarrow{LOS_{p}} \times \overrightarrow{V_g} \right\|}{\left\| {}^{I}\overrightarrow{LOS_{p}} \right\| \left\| \overrightarrow{V_g} \right\|} \tag{21} $$

3. Kinematic Equations

$$ \dot{\eta} = -\frac{\left\| \overrightarrow{V_g} \right\|}{\rho}\cos\eta + {}^{I}\dot{\psi}_{B} \tag{22} $$

$$ \dot{\rho} = -\left\| \overrightarrow{V_g} \right\| \sin\eta \tag{23} $$

$$ \dot{\lambda} = \frac{\left\| \overrightarrow{V_g} \right\|}{\rho}\cos\eta \tag{24} $$

$$ \dot{\varepsilon}_{\psi} = \left(-K_{pg}\varepsilon_{\psi} + \frac{\left\| \overrightarrow{V_g} \right\|}{\rho}\cos\eta - {}^{I}\dot{\psi}_{B} \right) \tag{25} $$

4. Derivation of Kinematic Equations

With the angles as they are defined, the kinematic equations are fairly simple to derive. Simple knowledge of angular rates allows for the definition of $\dot{\lambda}$ and $\dot{\rho}$.

$$\dot{\rho} = -\left\|\overrightarrow{V_g}\right\| \sin\eta \tag{26}$$

$$\dot{\lambda} = \frac{\left\|\overrightarrow{V_g}\right\|}{\rho} \cos\eta \tag{27}$$

From Figure 14 we can see that

$$\eta = -\lambda + {}^{I}\psi_B \pm 90° \tag{28}$$

Therefore taking the derivative we get

$$\dot{\eta} = -\dot{\lambda} + {}^{I}\dot{\psi}_B \tag{29}$$

Which simplifies to

$$\dot{\eta} = -\frac{\left\|\overrightarrow{V_g}\right\|}{\rho} \cos\eta + {}^{I}\dot{\psi}_B \tag{30}$$

From Figure 14,

$$\varepsilon_\psi = \lambda - {}^{I}\psi_B - {}^{B}\psi_C \tag{31}$$

Therefore taking the derivative we get

$$\dot{\varepsilon}_\psi = \dot{\lambda} - {}^{I}\dot{\psi}_B - {}^{B}\dot{\psi}_C \tag{32}$$

Therefore from Figure 15 as well as earlier definitions

$$\dot{\varepsilon}_\psi = \left(-K_{pg}\varepsilon_\psi + \frac{\left\|\overrightarrow{V_g}\right\|}{\rho} \cos\eta - {}^{I}\dot{\psi}_B \right) \tag{33}$$

5. Assumptions

The control laws developed from 2-dimensional kinematics. The aircraft model that is used for this project has a vertical channel within its autopilot that will maintain

constant altitude. Therefore there is no need to consider the kinematics of the aircraft in the vertical direction because they are negligible.

An assumption has been made that the error angle ε_ψ is correctly output from the image processing software, even though there will be errors in both ε_ψ and ε_θ. In normal flight of an aircraft there is usually some angle of attack that must be held in order to maintain altitude (Ref. 2). Assuming that the aircraft is positioned at some small angle of attack during flight, this will induce roll in the camera when it is not directly aligned with the longitudinal axis of the aircraft. With this roll present there will be some small error between the error angles found from the output of the image processor and the actual error angles present in the 2-dimensional problem. The image processor outputs the values u and v, however these can easily be transformed into the values of ε_ψ and ε_θ using the equations below.

$$\varepsilon_\theta = \tan^{-1}\left(\frac{v}{f}\right) \tag{34}$$

$$\varepsilon_\psi = \tan^{-1}\left(\frac{u}{f}\right) \tag{35}$$

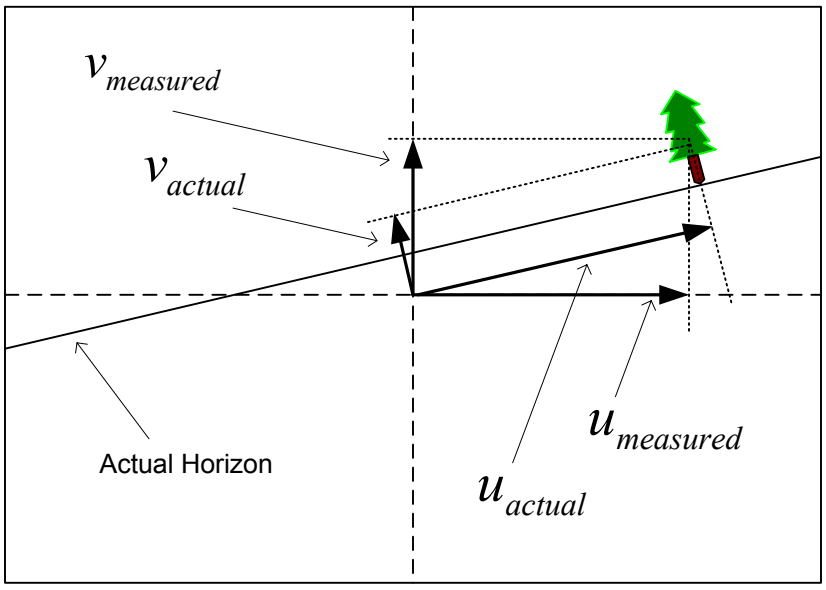

Figure 16. Rationale for 2 dimensional assumption even with presence of "roll" in camera

C. DEVELOPMENT OF CONTROL LAWS

In development of the control laws, the turn rate of the aircraft was selected to be the output of the control. The autopilot of the UAV was able to take the turn rate commands and turn them into corresponding aileron and rudder commands. The next decision was what method to use in controlling the turn rate, and this is the topic of the next few subsections.

There are many ideas that could be used in order to allow a UAV to track a stationary target. For a vehicle equipped with a movable camera, the most logical solution was to have the aircraft do circles around the target. It was also decided that it would be desirable to also have control over the radius of the circle for tactical reasons. For example, this would allow the user to move the UAV closer to the target to get a better view and then move farther away in order to remain unnoticed.

In order to accomplish the goals of this project, it was necessary to create two separate control laws that the UAV would switch between when applicable. Until a target is selected, both the aircraft and the camera must be controlled by an operator. Once a target is designated by the operator, the first control law can be activated. The first control law simply drives the aircraft's trajectory toward a circle around the target. While following the circular guidance law, an estimate of the range to the target is created. This is necessary because the range is not directly measured and must be estimated by methods developed in chapter VI of this thesis. The second control law compares the estimated range value to the desired range specified by the user to drive the trajectory to a circle around the target with a radius equal to the desired range.

1. Circular Guidance Law

Once the target is selected and the circular guidance law is activated, the turn rate command sent to the aircraft is designed to null the value of η. If the target is to the right of the aircraft velocity vector at the time of activation, then the control law will drive the trajectory to a clockwise circle. Otherwise, the trajectory will be that of a counterclockwise circle.

Once the turn direction has been determined, the turn rate commands are created through the use of a PI controller. The gains for the controller are -.5 and -.03.

$$\dot{\psi}_c = -.5\eta - .03\int \eta \tag{36}$$

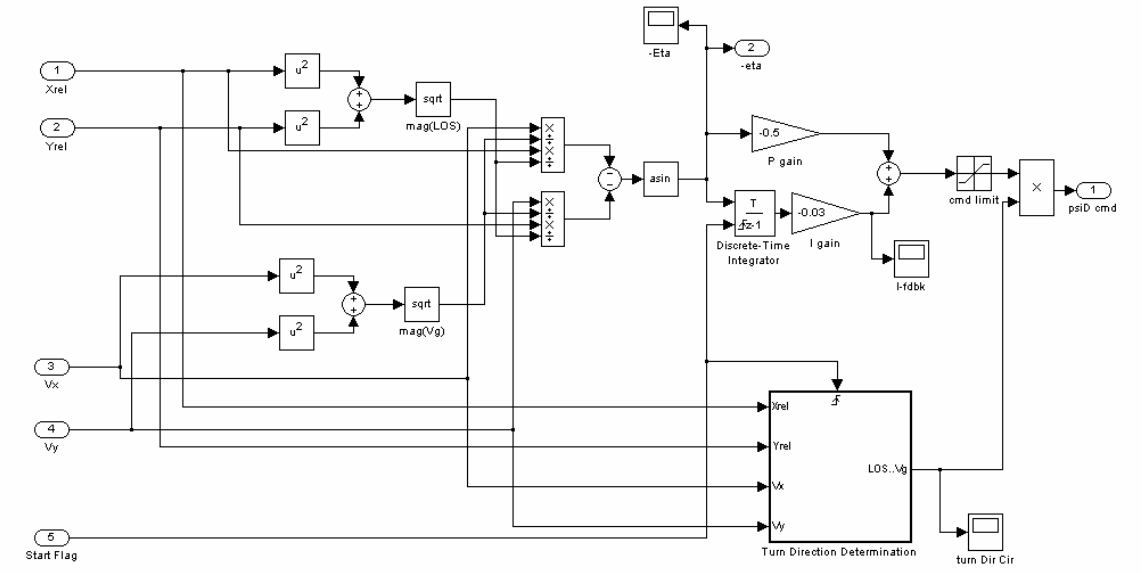

Figure 17. Illustration of Circular Guidance Law

2. Range Control Guidance Law

Once an accurate measurement of the range has been obtained, the second guidance law can be activated to drive the trajectory to that of a circle around the target with the desired range as the radius. The turn direction that was found in the first control law is still used in this case. The control law then compares the desired range to the actual range to obtain the range eroor, $\Delta\rho$, and both η and $\Delta\rho$ are used to determine the necessary turn rate.

As with the circular guidance control law, it was important to make the control law as simple as possible so that it could be easily employed in an actual system. It was also important that it be easily tunable to correspond to the aircraft that it was employed in. For this reason, a proportional plus integral (PI) control law was used again. In this case, PI controllers were constructed for both η and $\Delta\rho$ with a limiter applied to $\Delta\rho$ as a method of preventing windup of the integrator (Ref. 4).

$$\dot{\psi}_c = -.5\eta - .03\int \eta + .002\Delta\rho + .0002\int \Delta\rho \tag{37}$$

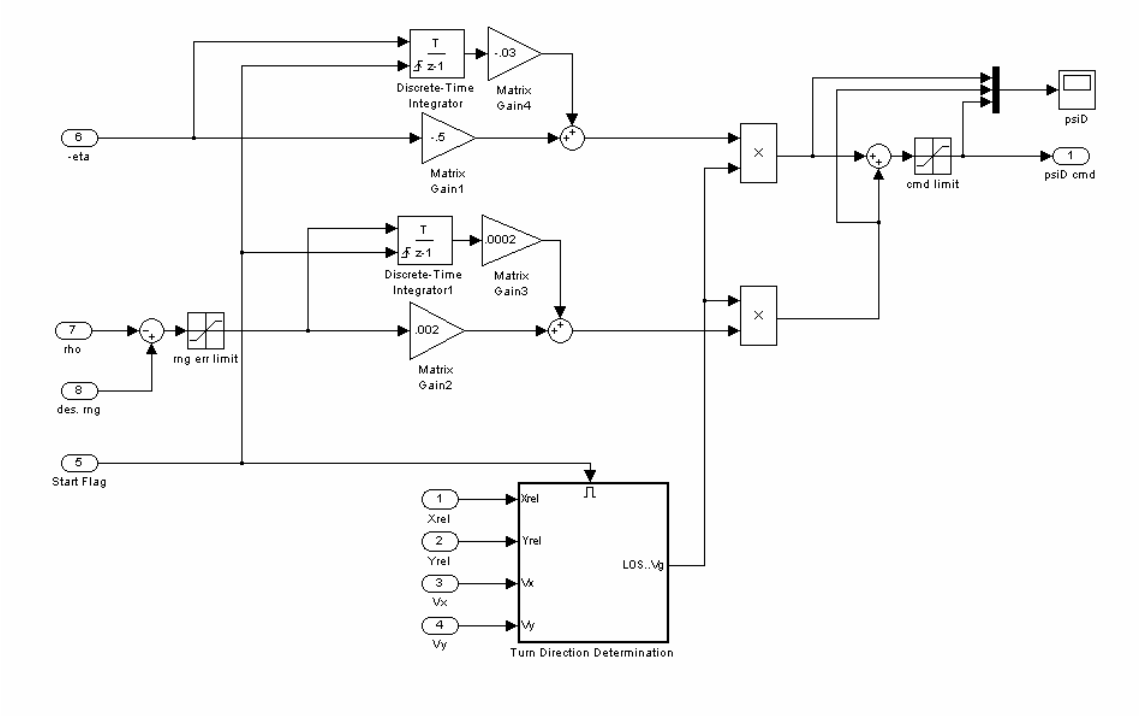

Figure 18. Illustration of Range Control Guidance Law

D. TESTING OF CONTROL LAWS USING SIMPLE KINEMATICS MODEL

In order to test the control laws, the simple kinematics equations for $\dot{\eta}$ and $\dot{\rho}$ were modeled with the two control laws that have been developed. The equations are presented here again for reference.

$$\dot{\eta} = -\frac{\left\|\overrightarrow{V_g}\right\|}{\rho}\cos\eta + {}^{I}\dot{\psi}_{B} \tag{38}$$

$$\dot{\rho} = -\left\|\overrightarrow{V_g}\right\|\sin\eta \tag{39}$$

1. Circular Guidance Law

The circular guidance law was simulated with initial conditions representative of the operational scenario. The initial value of η was set to 90° because it would be expected that the camera is looking forward to acquire the target.

a. Initial Conditions

- $\eta = 90^{\circ}$
- ρ = 500 m
- $\left\|\overrightarrow{V_g}\right\|$ = 50 m/s

b. Results

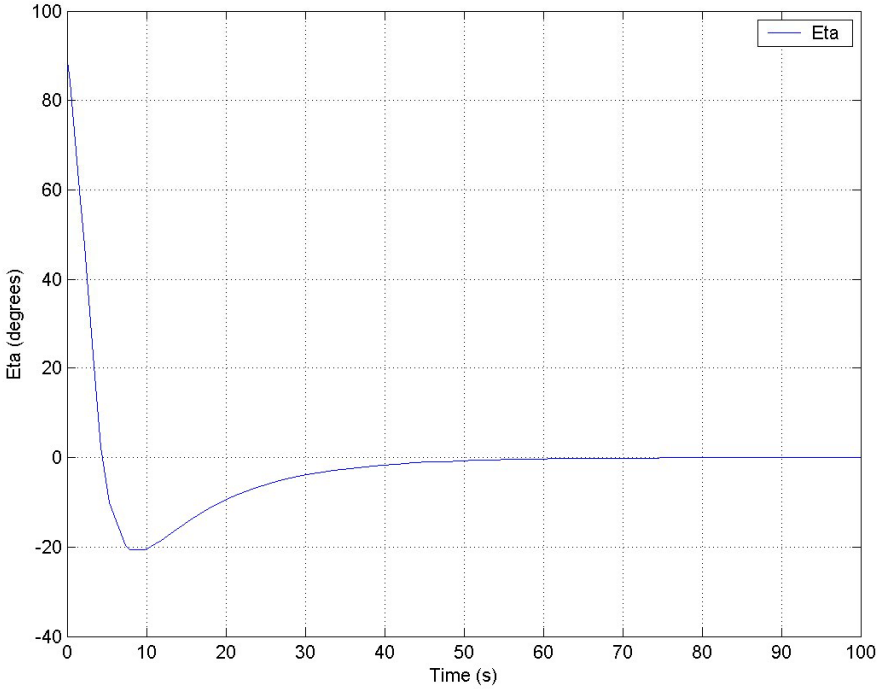

Figure 19. Eta vs. Time for circular control law testing

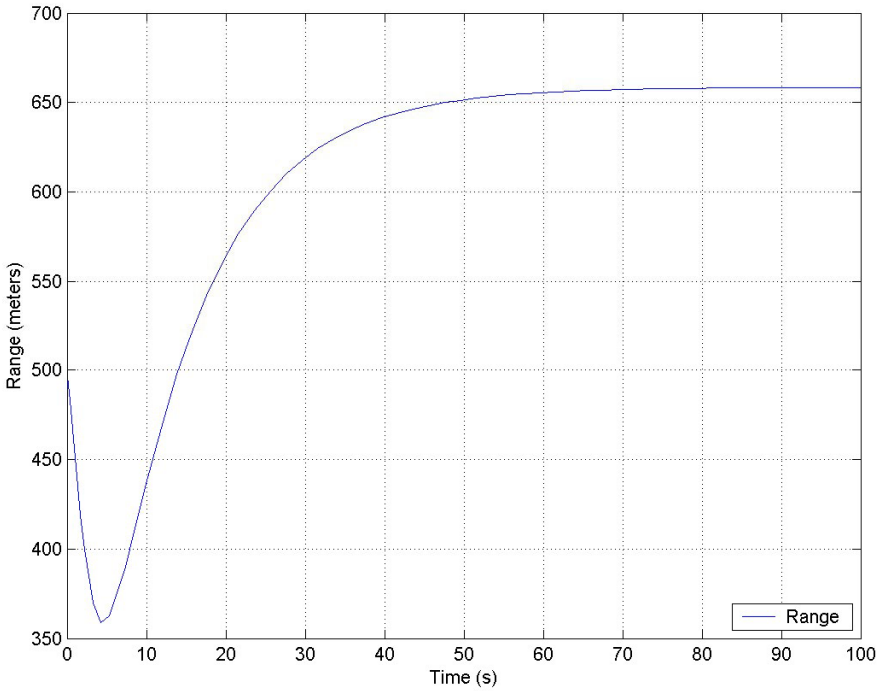

Figure 20. Range vs. Time for circular control law testing

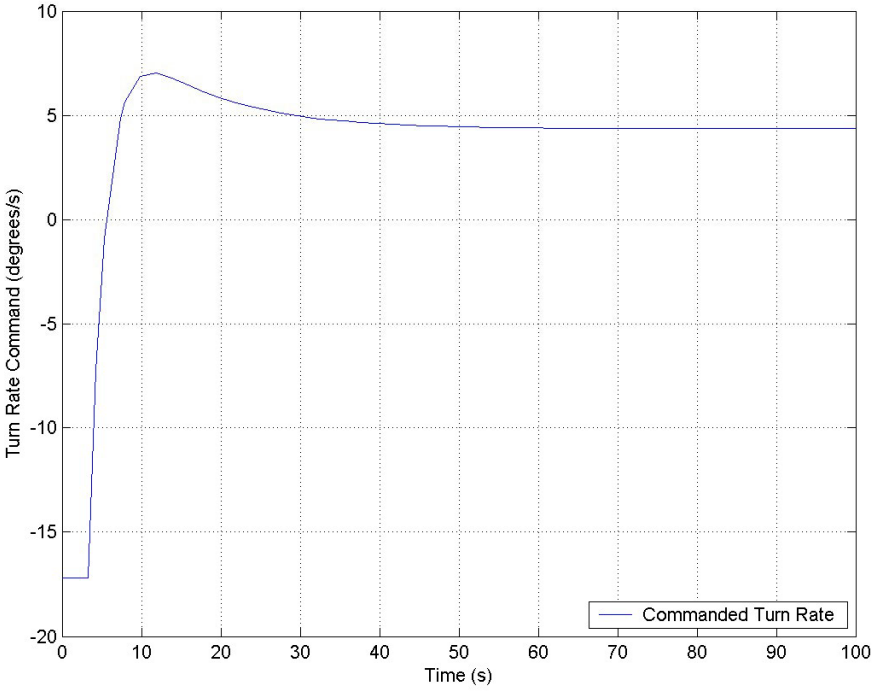

Figure 21. Commanded Turn Rate vs. Time for circular control law testing

These results show that the initial guidance law works well. It is able to quickly drive η to zero with only a small amount of overshoot. It also drives both the radius and turn rate commands to steady state values.

2. Range Control Guidance Law

The range control guidance law was simulated with initial conditions representative of what would occur in actual use. The initial value of η was set to 0° because it is expected that the range control guidance law would only be activated once the circular guidance law has been active for a long enough period of time to reach steady state. The desired range, $\overline{\rho}$ was set to 300 m so that the UAV would have to close toward the target to decrease range.

a. Initial Conditions

- $\eta = 0^{\circ}$
- $\rho = 500$ m
- $\overline{\rho} = 300$ m
- $\left\|\overrightarrow{V_g}\right\| = 20$ m/s

b. Results

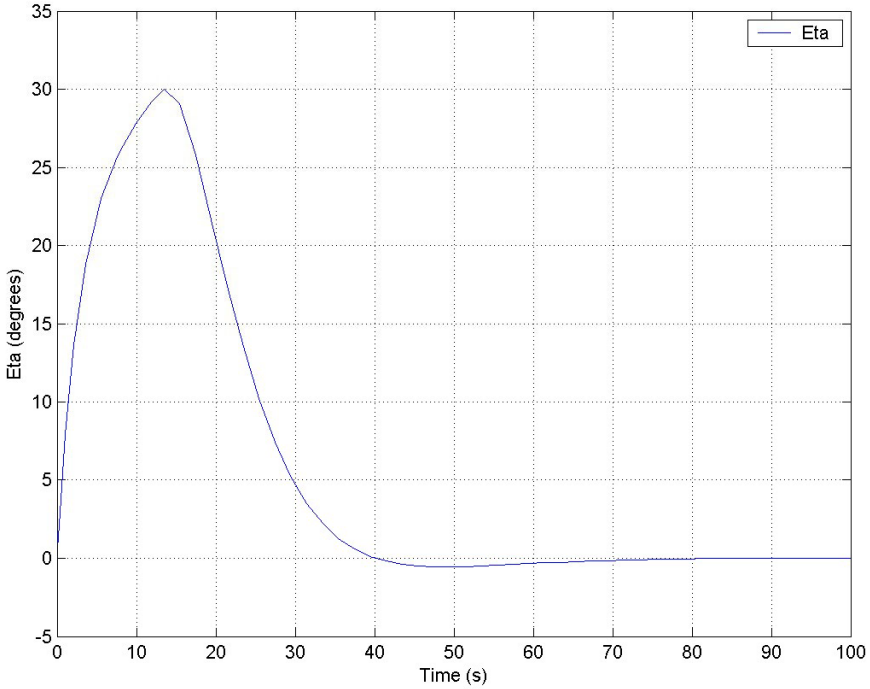

Figure 22. Eta vs. Time for Range Control law testing

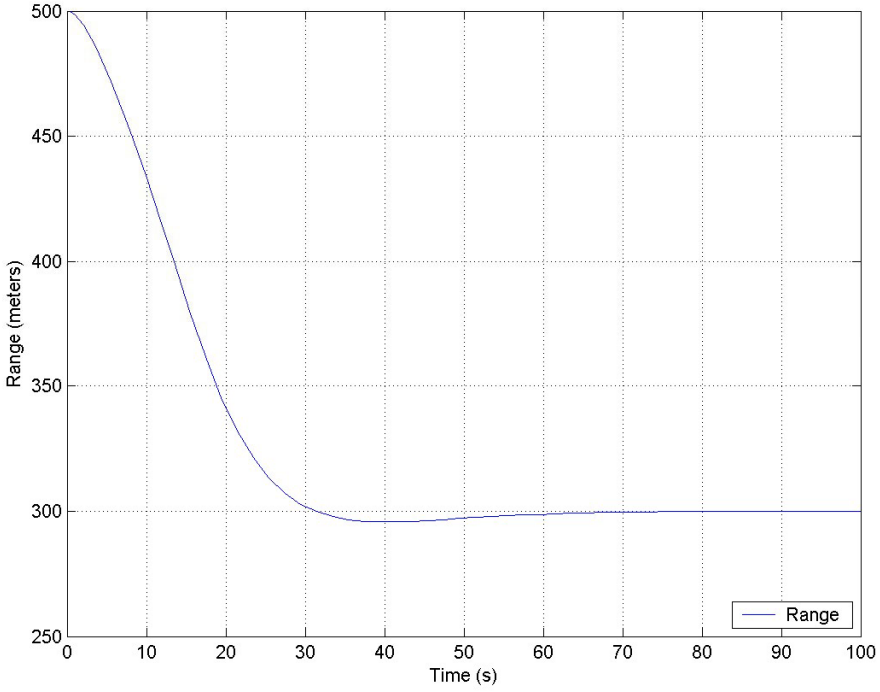

Figure 23. Range vs. Time for range control guidance testing

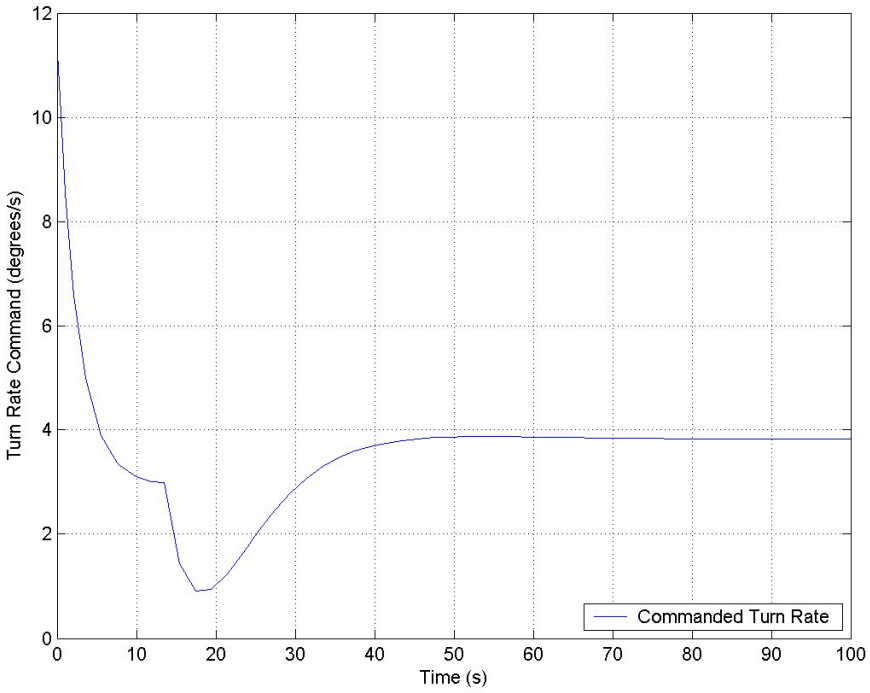

Figure 24. Commanded Turn Rate vs. Time for range control guidance testing

The results of this simulation show that the range control guidance law works well. The range is quickly reduced to its desired value, while η is increased in order to allow this range transition to occur. Once the desired range is reached, η returns to zero, and both the commanded turn rate and range reach a steady state value.

E. APPLICATION TO ACTUAL DYNAMIC SYSTEM

In order to apply the control laws developed here to the actual system, it is necessary to compute the vectors and angles that they use as inputs. First of all, the line of sight vector must be determined, and then this vector, along with the other known inputs from the system, can be used to determine the value of η.

1. Line of Sight Vector Determination

Determination of the line of sight vector is a simple matter of transforming the angular position of the target in the camera frame back to the inertial frame.

$$^{I}\overrightarrow{LOS} = {}^{I}_{C}C\,{}^{C}\overrightarrow{LOS} \tag{40}$$

Where the value of ${}^{C}\overrightarrow{LOS}$ is a vector determined from the three dimensions found by the camera and image processor; focal length, error in u-direction, error in v-direction.

$$ {}^{C}\overrightarrow{LOS} = \begin{bmatrix} f \\ u \\ -v \end{bmatrix} \tag{41} $$

With this vector known, it is a simple matter of trigonometry to determine the LOS angle λ.

If the camera is directly pointed at the target at all times, then this computation can be further simplified to be

$$ {}^{I}\overrightarrow{LOS} = {}^{I}_{C}C \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \tag{42} $$

This may lead to slightly more error and slower response, however it simplifies the control system which may make the entire system work better.

2. Determination of Eta Angle

In order to implement both of the control laws, the angle η must be known as it is directly used by both controllers. The angle η represents the angle between two 2-dimensional vectors because the guidance law that was developed is based on a 2-dimensional assumption. Therefore, the vectors in the following equations have been simplified to 2-dimensions, simply ignoring the z-component. There are a variety of ways that η can be computed, however the method shown below was selected for its simplicity.

$$ \left\| {}^{I}\overrightarrow{LOS}_{p} \times \overrightarrow{V_g} \right\| = \left\| {}^{I}\overrightarrow{LOS}_{p} \right\| \left\| \overrightarrow{V_g} \right\| \sin(\eta) \tag{43} $$

$$ \eta = \sin^{-1} \frac{\left\| {}^{I}\overrightarrow{LOS_{p}} \times \overrightarrow{V_g} \right\|}{\left\| {}^{I}\overrightarrow{LOS_{p}} \right\| \left\| \overrightarrow{V_g} \right\|} \tag{44} $$

THIS PAGE INTENTIONALLY LEFT BLANK

VI. RANGE ESTIMATION

A. OVERVIEW

In the development of the control system, it was determined that the range to the target should be controllable by a user input. In order to control a state of a dynamic system, that state must either be known or it must be estimated using available measurements. Once a target is designated, the only information known about it is its position within the image plane reference frame. Since this is only a 2 dimensional reference frame, it determines the line that the target must lie on, but gives no information about where on the line the target is. Therefore the range to the target must be estimated because it can not be directly inferred from the measurements of any of the sensors.

A variety of methods for range estimation were investigated for this problem. The circular trajectory around the target allowed for sufficient data to be collected, and the only decision was what method to use for actual estimation. Both triangulation and kalman filters based on knowledge of the problem geometry were examined. These methods will be discussed in the following sections and some conclusions about their effectiveness will be made.

B. TRIANGULATION

Triangulation is simply determining the intersection of two lines. With the position of the UAV known at two different times, as well as the angles from those positions to the target, then it is possible to estimate the position of the target as the point where those two vectors cross. With the positions of the aircraft and the line of sight vector to the target known at two separate instances in time, the following equations are derived.

$$\begin{aligned} x_1 + \rho_1 \cos\lambda_1 &= x_2 + \rho_2 \cos\lambda_2 \\ y_1 + \rho_1 \sin\lambda_1 &= y_2 + \rho_2 \sin\lambda_2 \end{aligned} \tag{45}$$

where x_1 is the x position in the inertial frame at time 1, y_1 is the y position in the inertial frame at time 1, λ_1 is the line of sight vector with respect to the inertial frame at time 1, and x_2, y_2, and λ_2 all correspond to time 2.

This equation can be rewritten in the following form which determines the current and previous range given all the necessary information.

$$\begin{bmatrix} \rho_1 \\ \rho_2 \end{bmatrix} = \begin{bmatrix} -\cos\lambda_1 & \cos\lambda_2 \\ -\sin\lambda_1 & \sin\lambda_2 \end{bmatrix}^{-1} \begin{bmatrix} x_1 & x_2 \\ y_1 & y_2 \end{bmatrix} \begin{bmatrix} 1 \\ -1 \end{bmatrix} \tag{46}$$

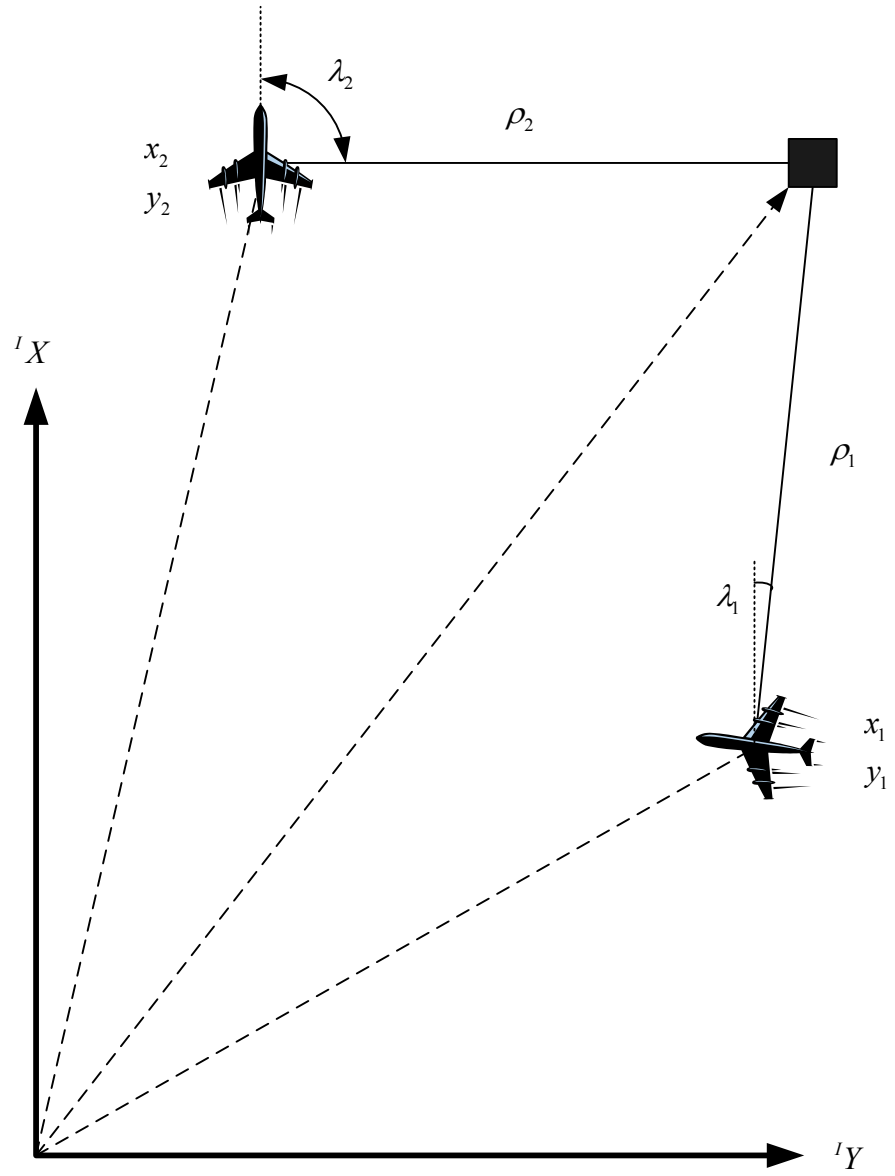

Figure 25. Overview of Triangulation Problem

To implement the equations above requires that the data be stored for a finite period of time, and then used along with current data to compute both the prior and current ranges. This requires a substantial amount of memory and computational power in order to carry out the calculations in real time. Once a data set is used as the old values in the current range calculation, it can be discarded so there is a constant flow of data into and out of the memory of the computer.

The challenge with any method of estimation is to ensure accuracy even in the presence of measurement noise. It is assumed that there will be Gaussian white noise present in the measurements made for triangulation. In order to minimize the effect of noise on the accuracy of the range estimate, it is important to ensure that the LOS vectors used in triangulation calculations are nearly orthogonal, corresponding to a low dilution of precision (Ref. 1). The figure below shows how errors can become large if non-orthogonal vectors are used.

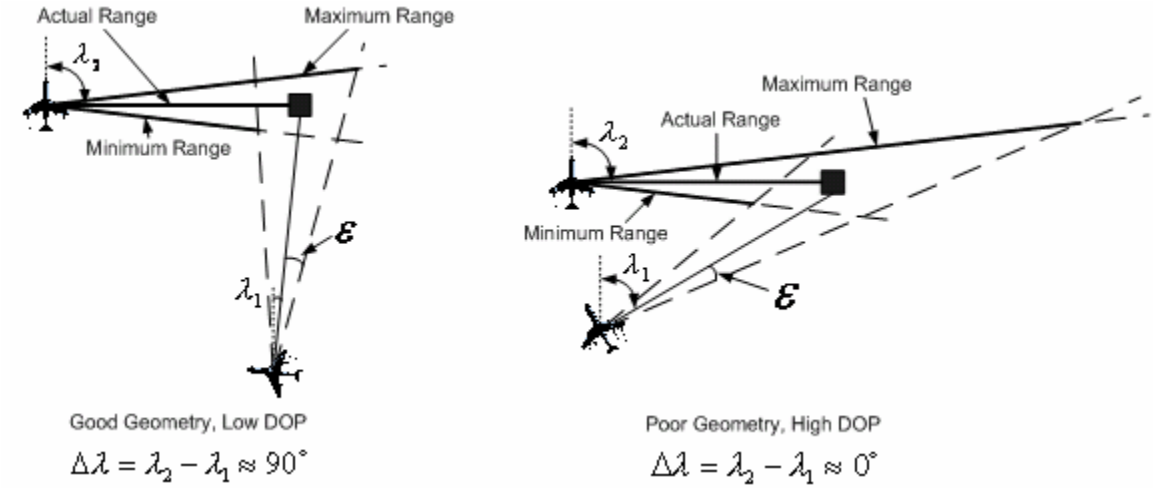

Figure 26. Range errors associated with noise in Triangulation Problem

As illustrated in the above figure, the orthogonality of the LOS vectors is important in getting accurate results. Defining $\Delta\lambda$ as the angular distance between measurements and ε as the angular error in measurement, the minimum and maximum range values that will be measured are described by the following equations

$$\rho_{\max} = \frac{\rho_0 \sin\left(\pi - \frac{\Delta\lambda}{2}\right)}{\sin\left(\frac{\Delta\lambda}{2} - \varepsilon\right)} \tag{47}$$

$$\rho_{\min} = \frac{\rho_0 \sin\left(\frac{\Delta\lambda}{2}\right)}{\sin\left(\pi - \frac{\Delta\lambda}{2} - \varepsilon\right)} \tag{48}$$

These equations show that in order to decrease the range uncertainty, it is necessary to either decrease the measurement uncertainty, ε, or make $\Delta\lambda$ close to 90°. As a further illustration of the range estimation errors, the maximum and minimum estimated ranges were computed based on different amounts of angular separation between data sets. For this computation, the actual range was set to be 500 ft and the angular error was set at 2°.

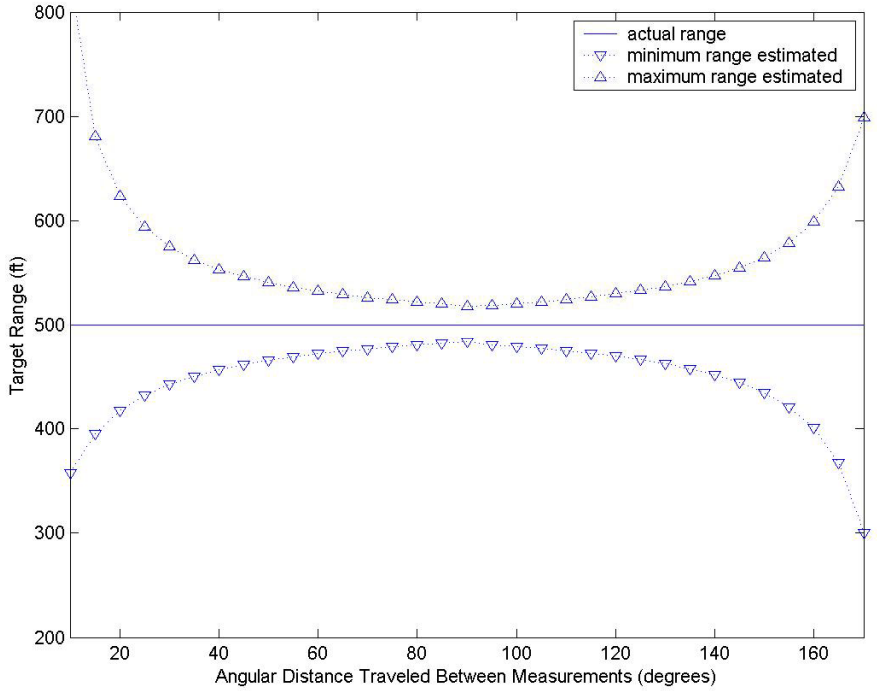

Figure 27. Illustration of maximum and minimum estimated ranges depending on the angular distance between measurements, assuming an angular error of 2° in the line of sight measurement and a range of 500 feet

The method for triangulation presented here should produce unbiased results with good accuracy as long as the measurement error is sufficiently small and data is collected from nearly orthogonal vectors. The data that is collected from noisy measurements, and the resulting range estimate is also noisy. Therefore, it is necessary to filter the information before it can be used within the control system.

The simplest method for filtering this data was to use a low-pass filter. This filtered out the high frequency oscillations of the signal and left behind a filtered range

estimate. The filter used here had a time constant of 12 seconds, and this clearly increases the transient response time of the estimate. More complicated filtering methods could be developed for more accurate results, however the current method provided the necessary insight to understand the effectiveness of triangulation as an estimation method.

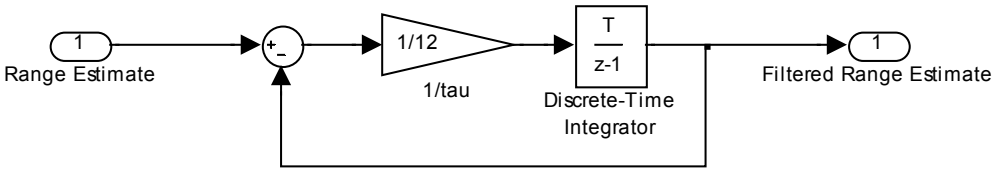

Figure 28. Diagram of Low-Pass Filter

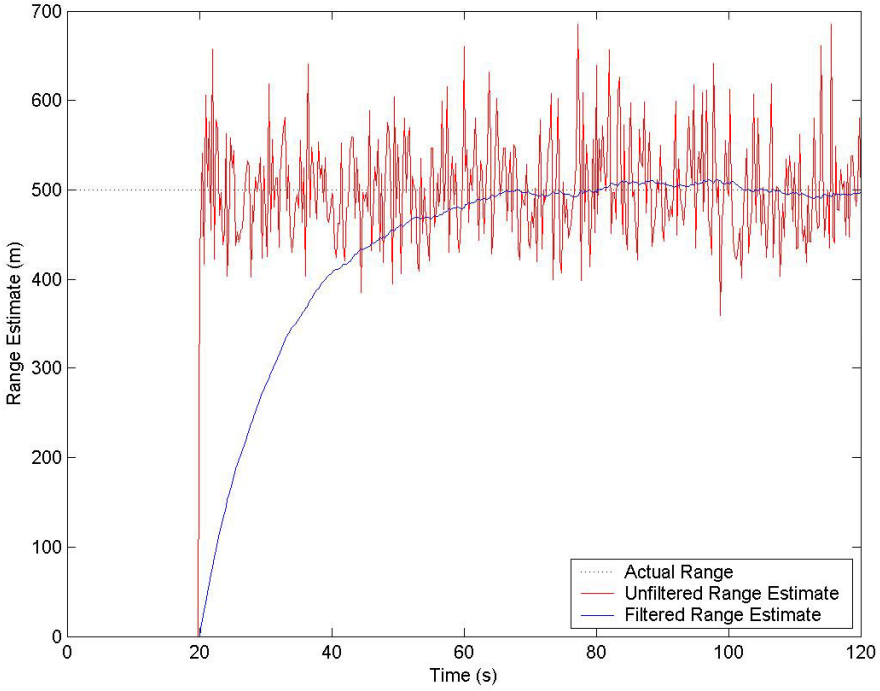

Figure 29. Filtered and Unfiltered Range Estimates using Triangulation

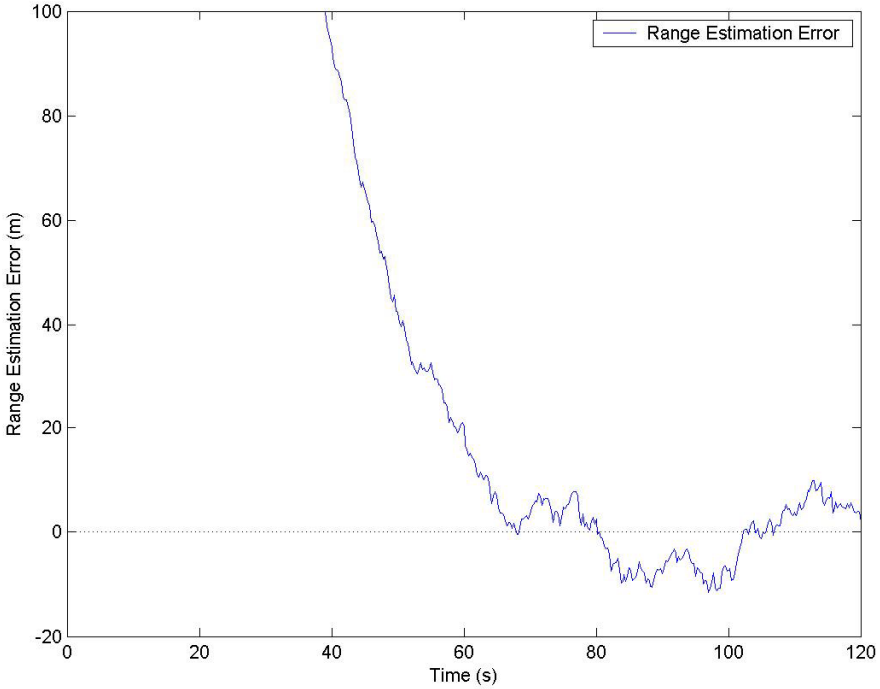

Figure 30. Range Estimation Error for Triangulation method with Low Pass Filter

The triangulation method gave good, unbiased results, however it was computationally difficult to implement. It required storing large quantities of data, retrieving data, processing it, and then discarding it. Beside accurate results, another benefit of triangulation is that an estimate of its accuracy can be made if the measurement noise is known.

C. RANGE ESTIMATE BASED ON LOS RATE FROM STEADY STATE KALMAN FILTER

1. Overview

This method of range estimation employed steady state Kalman filters to create estimates from noisy measurements. From the geometry of the problem, the following relationships could be developed.

$$v^p = \dot{\lambda}\rho = V_g \cos\eta \tag{49}$$

$$v^c = \dot{\rho} = -V_g \sin\eta \tag{50}$$

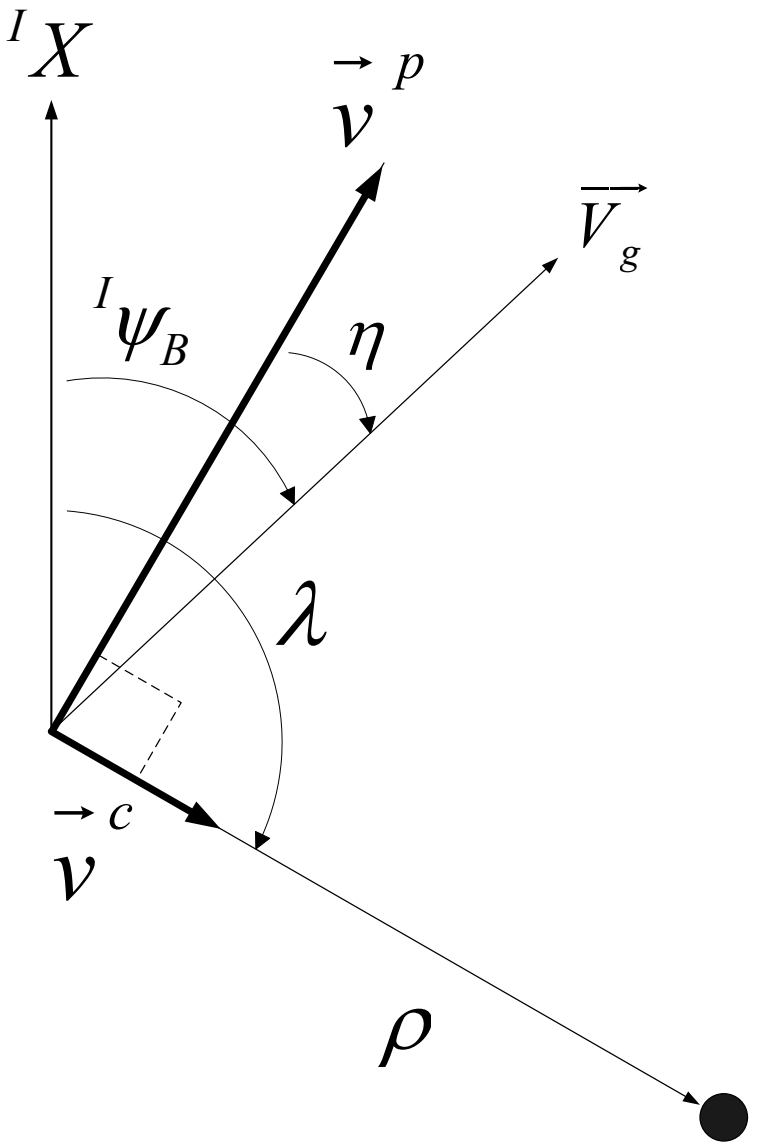

Figure 31. Geometry of Range estimation problem from LOS rate

Steady-state Kalman filters were used in this problem for a few reasons. As with control law development, it was desirable to use the simplest method that produces accurate results. A steady-state Kalman filter still gives good results at steady-state conditions and was acceptable for use here. It was also much simpler than the other types of filters that were examined for use on this problem. Furthermore, the residual obtained from a steady state filter is uncorrelated with the measurement errors, so an unbiased result can be obtained.

2. LOS Rate Estimation

The first step to this estimation problem is to create an estimate for the line-of-sight (LOS) rate of the target relative to the aircraft. The LOS angle, λ, is directly measurable and will be used along with an understanding of the system model to create an estimate of $\dot{\lambda}$.

a. System Equation

The system equation is given by

$$x_{k+1} = Ax_k + w_k \tag{51}$$

$$\text{where } x_{k+1} = \begin{bmatrix} \lambda_k \\ \dot{\lambda}_k \end{bmatrix},\ A = \begin{bmatrix} 1 & \Delta t \\ 0 & 1 \end{bmatrix},\ w_k \sim N(0, Q_k)$$

The value of x_{k+1} is the future update of the state vector, A is the state transition matrix defined by the dynamics of the system, and w_k is the process noise which is assumed to be a white, zero-mean Gaussian distribution with covariance matrix Q_k.

b. Measurement Equation

The measurement equation is given by

$$z_k = Cx_k + \mu_k \tag{52}$$

$$\text{where } C = \begin{bmatrix} 1 & 0 \end{bmatrix},\ \mu_k \sim N(0, R_k)$$

The value of z_k is the measurement vector at the current sample time, C is the measurement matrix that is determined from the observability of the state variables, and μ_k is the measurement noise which is assumed to be a white, zero-mean Gaussian distribution with covariance matrix R_k.

c. Kalman Filter

The Kalman filter equation is given by

$$\hat{x}_{k+1} = A\hat{x}_k + K_{k+1}(z_{k+1} - CA\hat{x}_k) \tag{53}$$

where K_{k+1} is the Kalman gain

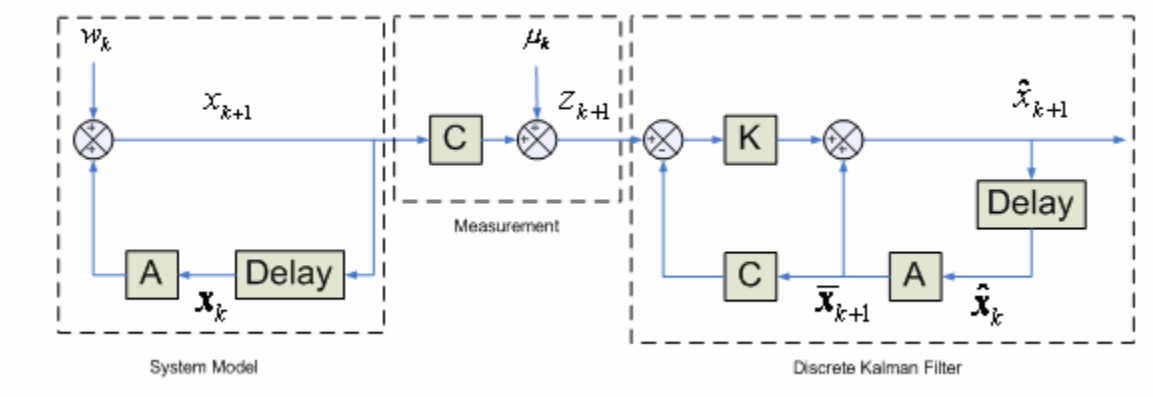

Figure 32. Diagram of Steady State Kalman Filter for Line of Sight Rate Estimation

d. Determination of Steady State Kalman Gains

The steady state Kalman gains were determined by running a model of a standard Kalman filter and using the steady state gains once they had converged. The steady state Kalman gain was selected to be

$$K = \begin{bmatrix} .3371 \\ .2716 \end{bmatrix}$$

The figure below shows the convergence of the individual gains within the standard Kalman filter. Both of the gains converge to their steady state values within a matter of seconds.

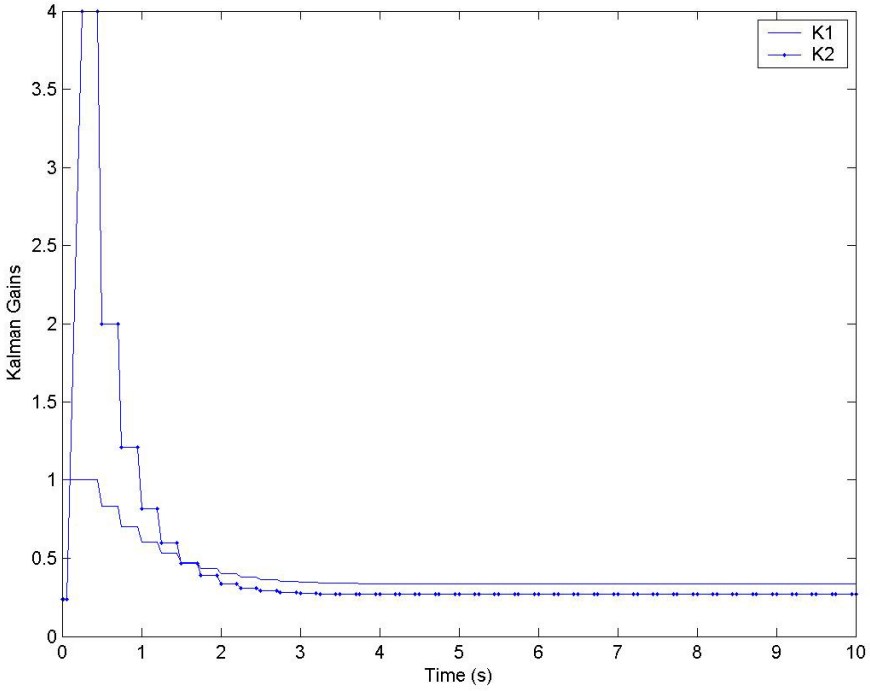

Figure 33. Kalman Gains for LOS rate Kalman Filter

The standard Kalman filter and the steady state Kalman filter were both applied to the same input signal to gain insight on their relative performance. Although the standard Kalman filter converges to the correct estimate much faster, both filters obtain the same steady state estimate. Therefore, it is possible to use a steady state Kalman filter and still obtain accurate results.

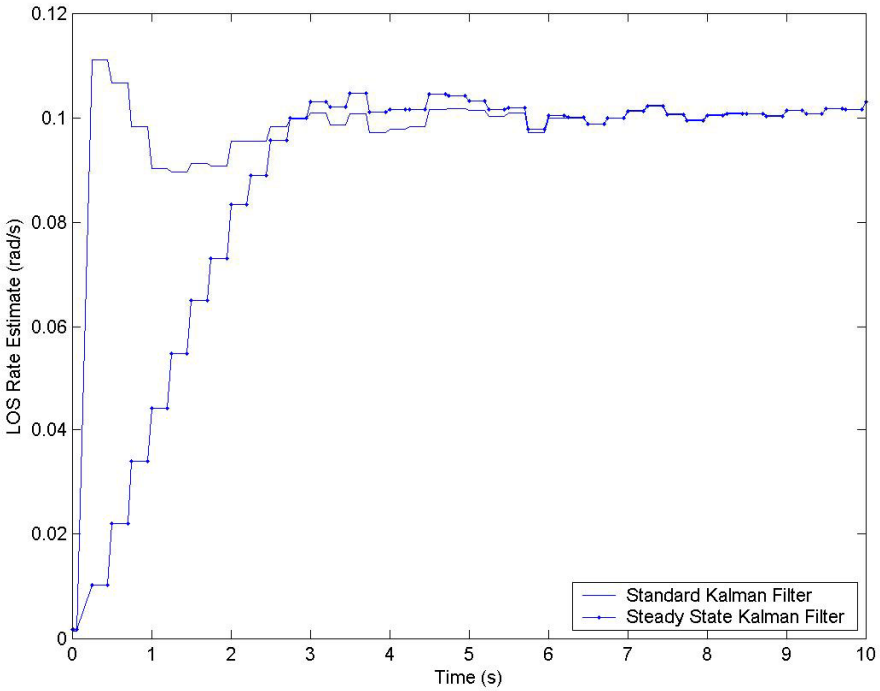

Figure 34. LOS Rate Estimates from Standard Kalman Filter and Steady State Kalman Filter

3. Range Estimate

With an estimate of the LOS rate, it is then possible to produce an estimate of the 2 dimensional range to the target. Again, a steady state Kalman filter was used for ease of implementation and its ability to produce an unbiased result even with measurement uncertainty.

a. System Equation

The system equation is defined by

$$x_{k+1} = Fx_k + w_k \tag{54}$$

$$\text{where } x_k = \begin{bmatrix} \rho_k \\ \dot{\rho}_k \end{bmatrix},\ F = \begin{bmatrix} 1 & \Delta t \\ 0 & 1 \end{bmatrix},\ w_k \sim N(0, Q_k)$$

The value of x_{k+1} is the future update of the state vector, F is the state transition matrix defined by the dynamics of the system, and w_k is the process noise

which is assumed to be a white, zero-mean Gaussian distribution with covariance matrix Q_k.

b. Measurement Equation

The measurement equation is defined by

$$z_k = H_k x_k + v_k \tag{55}$$

$$\text{where } z_k = \begin{bmatrix} v_k^p \\ \dot{\rho}_k \end{bmatrix},\ H_k = \begin{bmatrix} \hat{\dot{\lambda}}_k & 0 \\ 0 & 1 \end{bmatrix},\ v_k \sim N(0, R_k)$$

The value of z_k is the measurement vector at the current sample time, H_k is the measurement matrix that is determined from the geometry of the problem, and v_k is the measurement noise which is assumed to be a white, zero-mean Gaussian distribution with covariance matrix R_k.

c. Kalman Filter

The time update equation is given as

$$\bar{x}_{k+1} = F\hat{x}_k \tag{56}$$

The measurement update equation is given as

$$\hat{x}_{k+1} = \bar{x}_{k+1} + K_{k+1}(z_{k+1} - H_{k+1}\bar{x}_{k+1}) \tag{57}$$

$$\text{where } K_{k+1} = \bar{P}_{k+1} H_{k+1}'(R_{k+1} H_{k+1} \bar{P}_{k+1} H_{k+1}')^{-1}$$

$$\bar{P}_{k+1} = F P_k F' + Q_k$$

$$P_{k+1} = (I - K_{k+1} H_{k+1}) \bar{P}_{k+1}$$

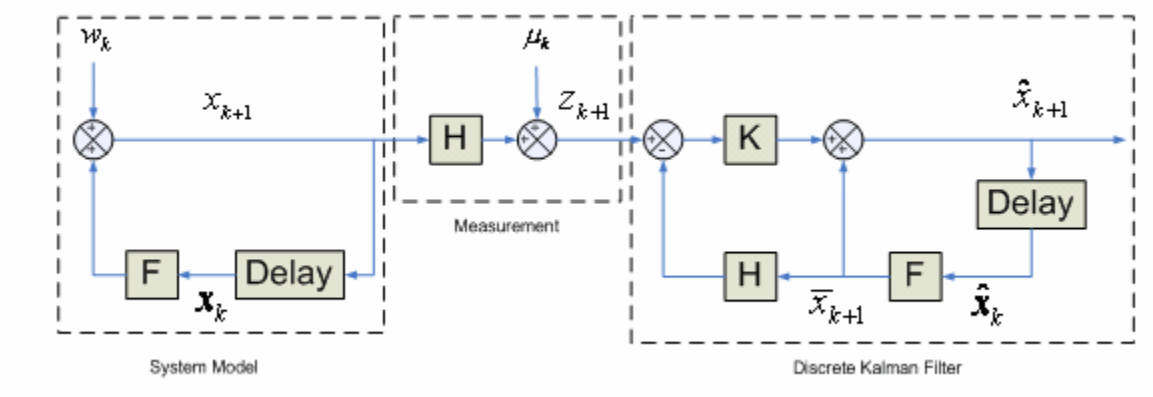

Figure 35. Diagram of Steady State Kalman Filter for Range Estimation

d. Determination of Steady State Kalman Gains

To determine the steady state gains for the range estimation Kalman filter, it was again necessary to run a simulation with the filter being applied to it and examine the individual Kalman gains once their steady state values have been reached. In order to do this, the filters were simulated with different values of $\dot{\lambda}$. The table below contains the steady state Kalman gains that were achieved with different LOS rates, and shows that there is very little dependence on the LOS rate, so it is possible to use a steady state Kalman filter for this problem.

LOS Rates [rad/s]	Steady State Kalman Gains
$\frac{20}{1500} = 0.0133$	$\begin{bmatrix} 0.0087 & 0.1415 \\ 0.0000 & 0.8679 \end{bmatrix}$
$\frac{20}{1000} = 0.02$	$\begin{bmatrix} 0.0087 & 0.1415 \\ 0.0000 & 0.8679 \end{bmatrix}$
$\frac{20}{500} = 0.04$	$\begin{bmatrix} 0.0087 & 0.1415 \\ 0.0000 & 0.8679 \end{bmatrix}$
$\frac{20}{200} = 0.1$	$\begin{bmatrix} 0.0087 & 0.1414 \\ 0.0000 & 0.8678 \end{bmatrix}$

Table 1. Comparison of Steady State Kalman Gains for different LOS rates showing that they are nearly independent of LOS rate

The Kalman gains for a filter with a LOS rate of .04 rad/s are shown in the figure below. This figure shows that most of the gains converge very rapidly to their steady state value. The value of K(1,1) converges very slowly compared to the rest, and modifications to the steady state Kalman filter had to be made in order to get good performance with this gain.

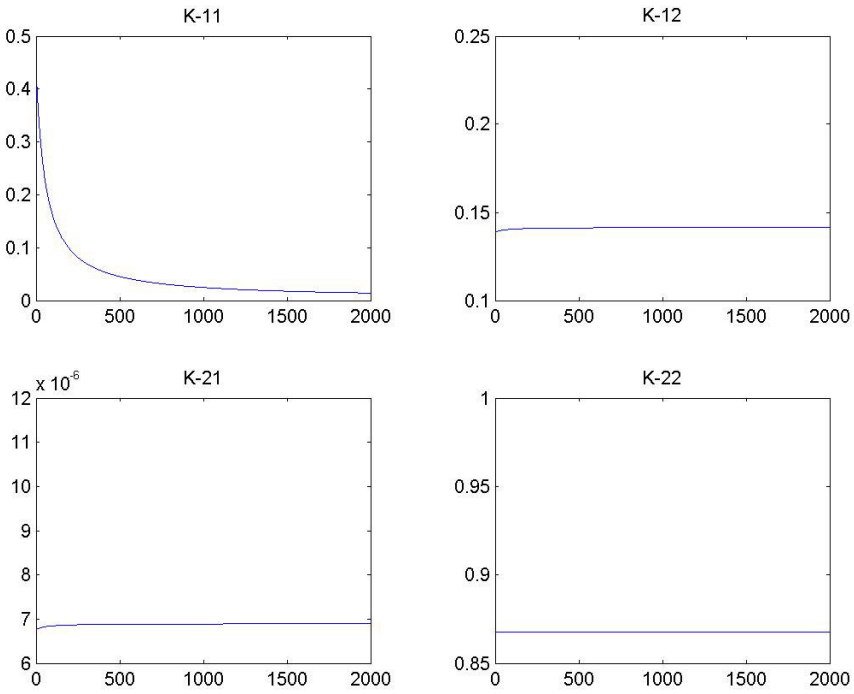

Figure 36. Kalman Gains for a filter with LOS rate of .04 rad/s

Because of the slow convergence for the K(1,1) element, it was necessary to modify that particular steady state gain for the Kalman filter. The following gains were used.

For the first 30 seconds:

$$K_{k+1} = \begin{bmatrix} 0.0087 \times 390 & 0.1415 \\ 0.0000 & 0.8679 \end{bmatrix}$$

For all time after 30 seconds:

$$K_{k+1} = \begin{bmatrix} 0.0087 \times 20 & 0.1415 \\ 0.0000 & 0.8679 \end{bmatrix}$$

The use of these gains allowed the range estimate to converge much faster than if the standard steady state gains were used. These speed up the transient response of the estimation at the cost of accuracy, however because the range can be constantly changing, it is necessary to speed up the transient response time.

e. Error Analysis

A steady state Kalman filter was used because of its simplicity as well as its ability to provide an unbiased result. Because the range estimate is based on an estimate of the LOS rate, the Kalman gain will become a function of the error in the LOS rate and will result in a biased range estimate.

The actual measurement equation should be

$$z_k = (H_k + \Delta H_k)x_k + v_k \tag{58}$$

where $H_k = \begin{bmatrix} \hat{\dot{\lambda}}_k & 0 \\ 0 & 1 \end{bmatrix}$ and $\Delta H_k = \begin{bmatrix} \dot{\lambda}_k - \hat{\dot{\lambda}}_k & 0 \\ 0 & 1 \end{bmatrix}$

The ΔH_k term represents the error between the measured and actual LOS rate. Looking at the error dynamics of these equations, it is possible to see that a bias will be present.

Defining

$$\Delta x_k = x_k - \hat{x}_k \tag{59}$$

We then get

$$E(\Delta x_{k+1}) = A(I - K_{k+1}H)FE(\Delta x_k) + E(K_{k+1}\Delta H_{k+1}Fx_k) \tag{60}$$

where E(*) represents the expectation values (Ref. 8)

$E(\Delta x_k) = 0$ so the first term will cancel out, however $E(K_{k+1}\Delta H_{k+1}Fx_k)$ will be non-zero because the Kalman gain K_{k+1} is a function of ΔH_{k+1}

D. SUMMARY

The range estimation techniques discussed here are both valid methods and produce good results. For ease of implementation, the steady state Kalman filter method was used in the actual control system that was developed. Triangulation is an accurate method however it is more computationally demanding. For real time implementation of this estimator it was important that it was a recursive solution that could easily run in real time. The accuracy that was achieved with this filter in initial testing was very good and should be satisfactory for actual implementation. Other methods can still be examined such as weighted least squares approximation, robust Kalman filter, and Krein Space filters, however these methods were currently overlooked because a simple steady state Kalman filter seemed to work sufficiently well.

THIS PAGE INTENTIONALLY LEFT BLANK

VII. SIMULATION AND TEST RESULTS

A. SIMULINK TESTING

The first method for testing the control system that was developed was to use a realistic Simulink model to simulate all aspects of the problem. This involved using the models of the PerceptiVU target tracking software, the pan-tilt unit, camera, and the UAV. Efforts were taken to make the models as realistic as possible, and to incorporate the appropriate sensor and measurement noise that can be expected in reality.

1. Simulink Model

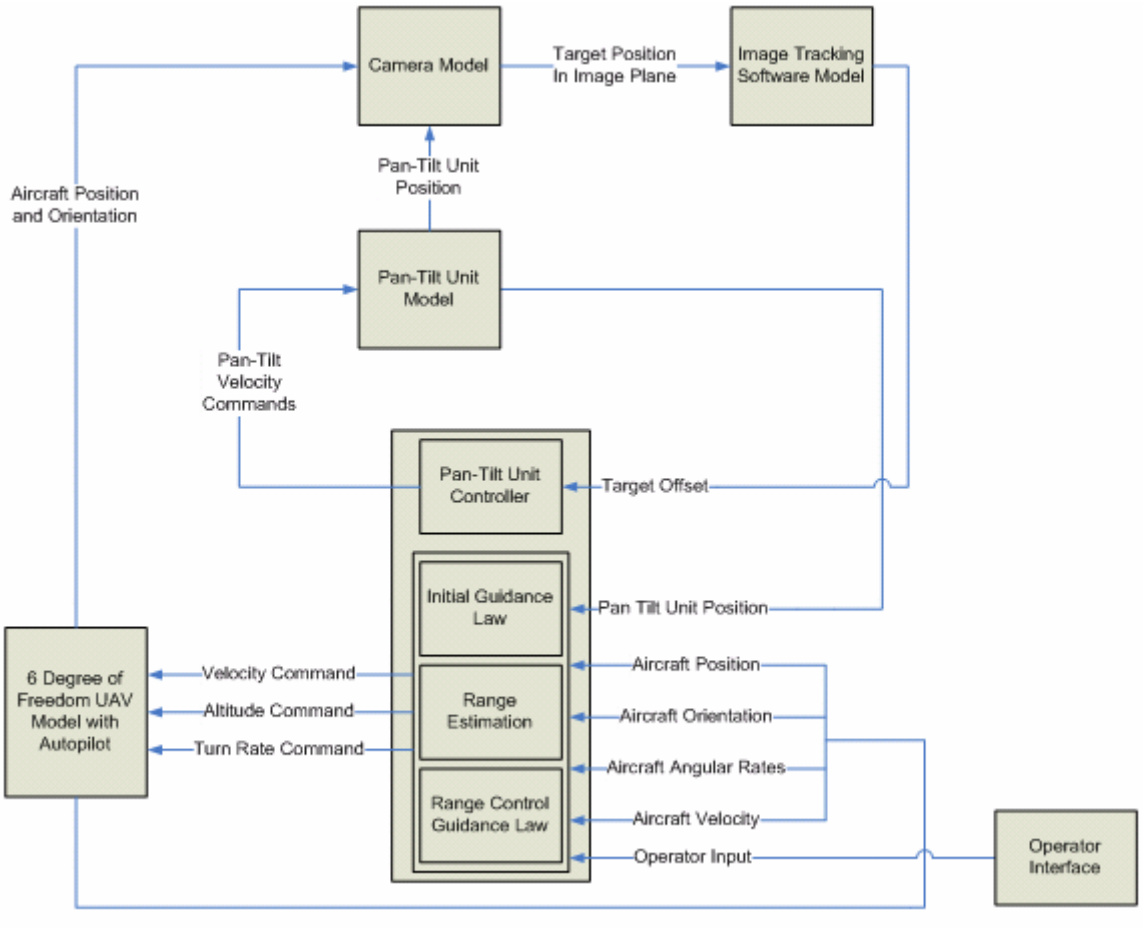

Figure 37. System Setup for Simulink Model Simulations

2. Test Conditions

A Simulink simulation was conducted with the following conditions:

Measurement Noise		
Noise Source	Mean	Standard Deviation
Camera boresight error	0	0.3 degrees
Camera Gimbal angle position	0	0.05 degrees
Inputs to Control System		
Input		Time
Start Initial Guidance Law		10 sec
Start Estimation of LOS Rate		20 sec
Start Range Estimation		30 sec
Begin Range Transition Guidance		100 sec
Desired Range is 400 ft		100 sec
Desired Range is 300 ft		200 sec
Desired Range is 200 ft		300 sec
Desired Range is 150 ft		400 sec
Desired Range is 100 ft		600 sec
Desired Range is 50 ft		800 sec

Table 2. Simulation Conditions for Simulink Simulation of Control System

3. Results

The following figures show the results of the simulation that was conducted. The horizontal trajectory of the aircraft around the target is shown in figure 38. It is clear that once the guidance laws have been initiated, the aircraft follows a circular trajectory unless it is undergoing a range transition.

Figure 39 shows the height of the aircraft during the simulation, which is commanded to be held at 100 meters. The height changes in response to maneuvers of the aircraft, however because of the height compensator that was created the height fluctuations are extremely small.

Figure 40 shows the transition between range commands. This shows that the aircraft responds quickly to range transition commands. Furthermore, it also shows that the range estimate remains accurate during maneuvers of the aircraft and is sufficiently fast to perform well during range transition commands.

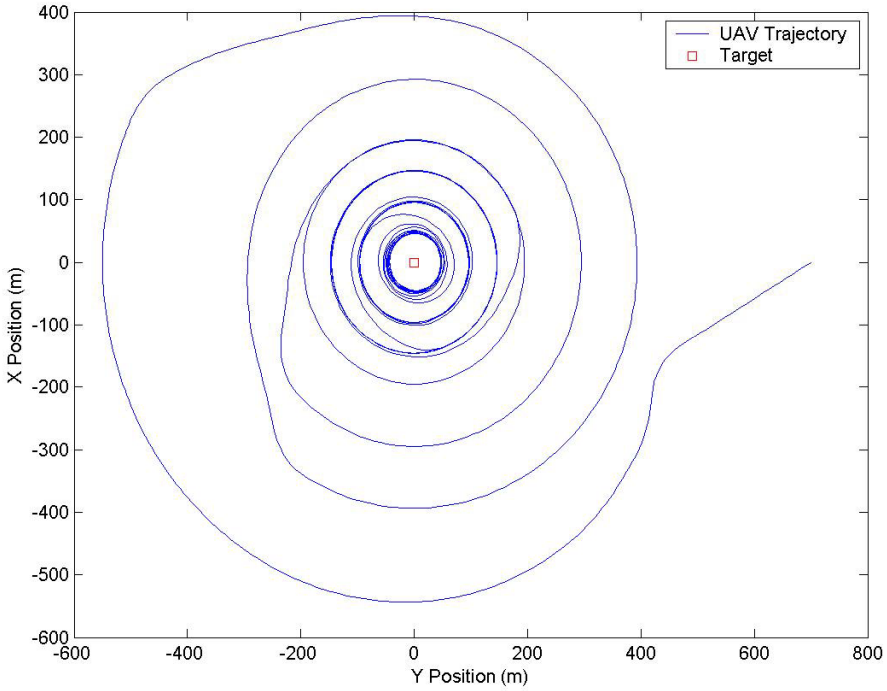

Figure 38. Horizontal Trajectory of UAV Around the Target

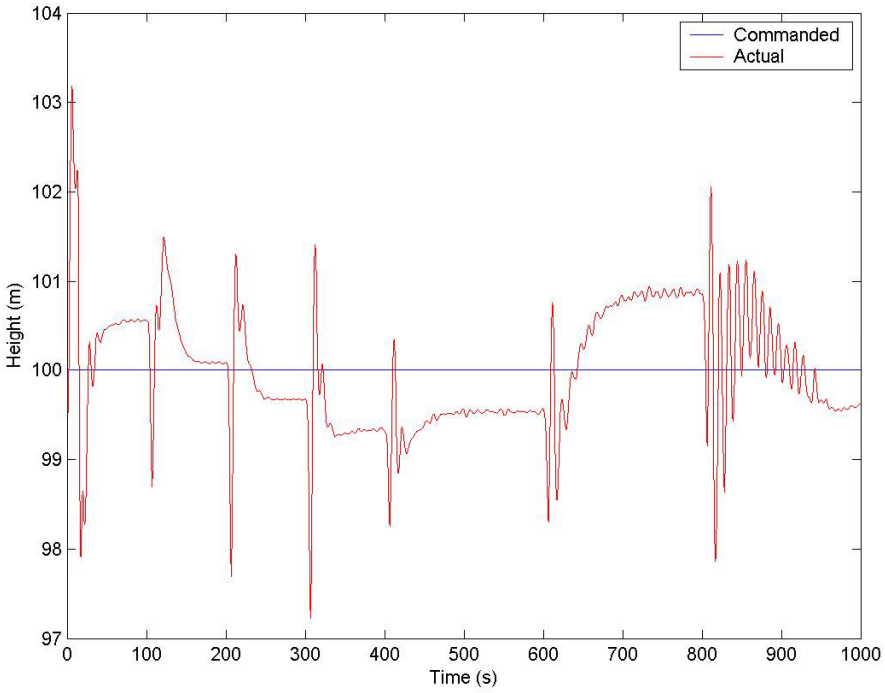

Figure 39. Commanded and Actual Height of UAV during Simulation

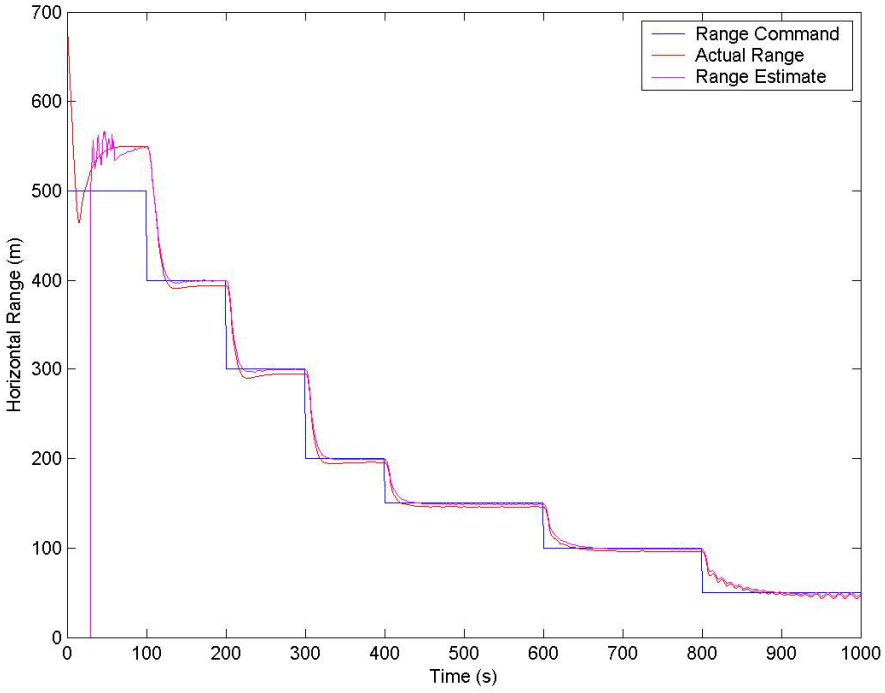

Figure 40. Commanded, Actual, and Estimated Ranges during Simulation

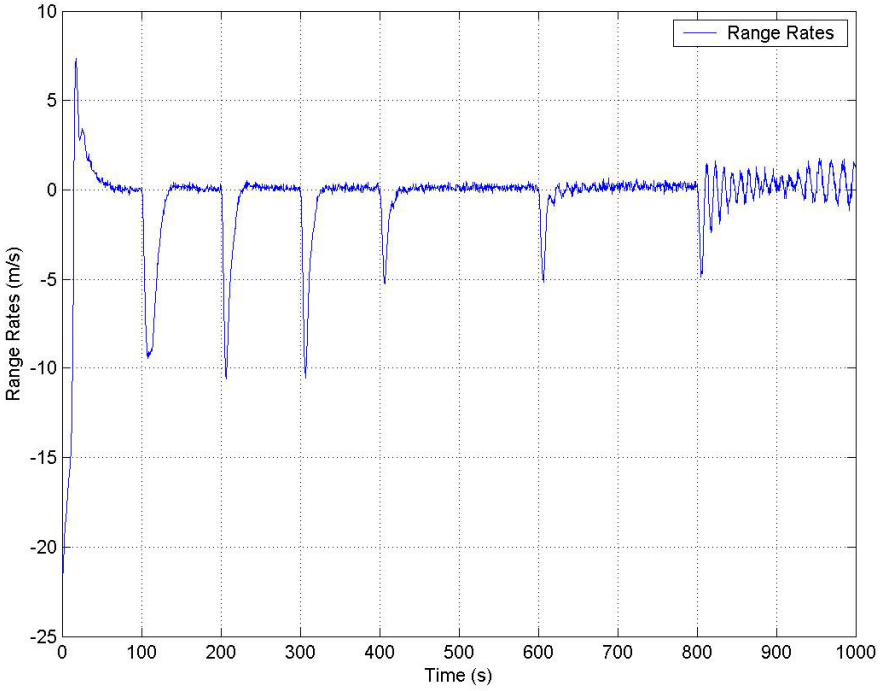

Figure 41. Horizontal Range Rates during Simulation

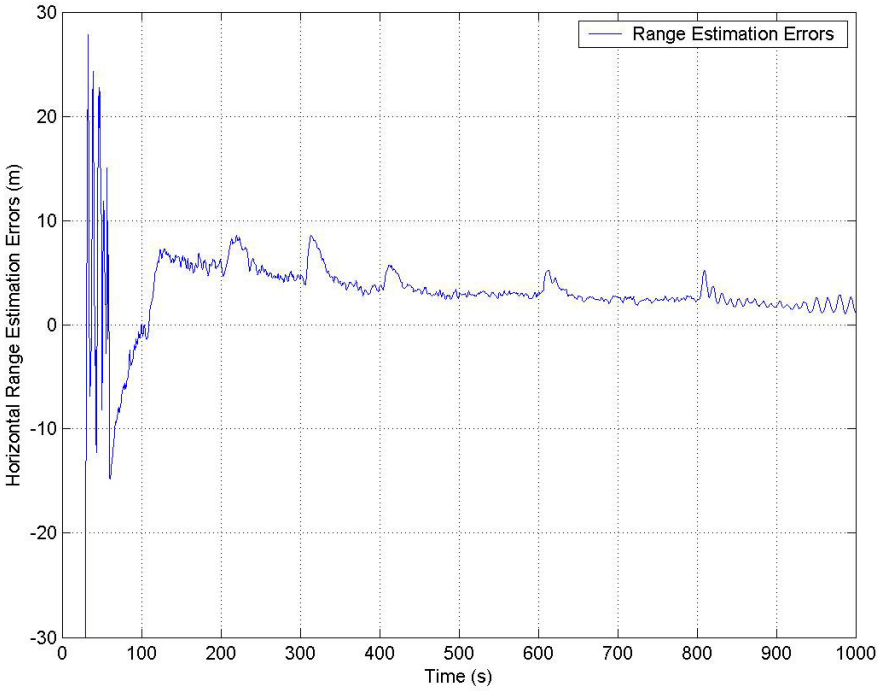

Figure 42. Horizontal Range Estimation Errors of the Steady State Kalman Filter Range Estimator during Simulation

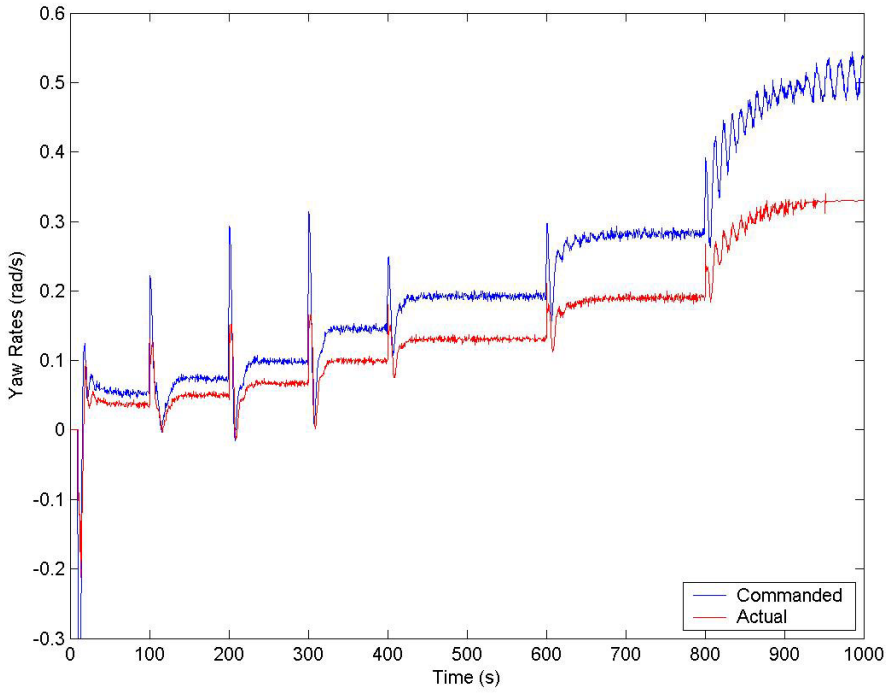

Figure 43. Commanded and Actual Yaw Rates during Simulation

In all, the results from the Simulink simulation prove that the control system developed for this project works very well. The control laws that have been developed appear to be very stable and the range estimator is capable of estimating the range to the target with a very small percentage of error.

B. TESTING OF PERCEPTIVU SOFTWARE

The PerceptiVU software is just one of the many components that must work together for autonomous tracking to work properly. It was purchased only after seeing demonstrations of its capabilities, but it was still necessary to ensure that it would be effective in tracking targets from a UAV mounted camera.

1. Video from Previously Recorded Flights

The first step in testing the software was to use video recordings from previous UAV flights with hard-mounted cameras and test the ability of the software to track targets. Attempts were made to track targets of different sizes and colors and resulted in varied success. The quality of the video and the stabilization of the image were also important factors for successful target tracking.

Although no quantitative results could be obtained from this testing, the following general observations were made:

- The target must be sufficiently large to occupy enough pixels on the screen for the software to be able to track it.
- The target must have sufficient contrast with the surrounding area. For example, one tree in a group of trees cannot be tracked, however a tree out in the open can easily be tracked if other conditions are met.
- The target must not move rapidly across the screen.
- Once the target leaves the field of view of the camera, it will no longer be tracked, even if it quickly re-enters.
- The software does not recognize that it has lost the target, and instead finds the next closest thing to track.

C. CLOSED LOOP TESTING WITH AVDS

After testing the control system within Simulink, the next step was to incorporate the majority of the hardware into the testing process. This included incorporating the PerceptiVU target tracking software and using the output to drive the control system. In order to incorporate the PerceptiVU target tracking software, it was necessary to create video images of the target in the camera frame. To do this, Aviator Visual Design Simulator (AVDS) was used. AVDS is a software program developed by RasSimTech Ltd. that interfaces with Simulink and provides a visual representation of an aircraft in flight. It allows for customizable terrain, so it was possible to add a target on the ground. Taking as input the position and orientation of the camera, it was possible to get a simulated camera view to output to the PerceptiVU target tracking software. The software is then able to determine the position of the tracked target within the image plane, and output the error values back to the control system.

1. Closed Loop System with AVDS

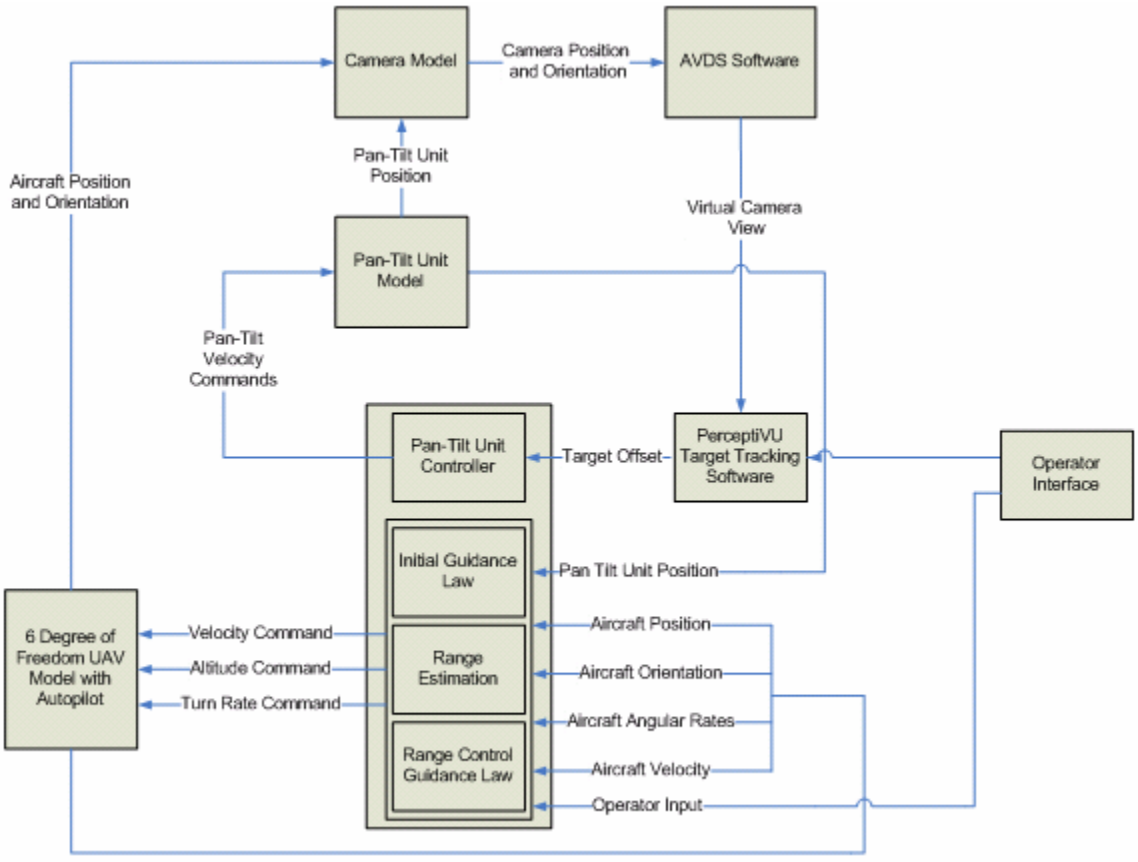

Figure 44. Setup for Hardware In Loop Simulation with AVDS and PerceptiVU

2. Test Conditions

Simulations were conducted for the hardware in the loop simulation. The initial conditions of the aircraft were set so that the target could be easily acquired. During target acquisition, the camera operator has control over the pan-tilt unit as well as the selection cue on the screen of the PerceptiVU software. Therefore the operator was able to move the camera manually until they could select the target and have PerceptiVU track the target automatically. Once the target was selected, the guidance control system could be activated that drove the aircrafts trajectory toward a circle around the target. The range estimator was also activated. Once the range estimator had converged, the range transition guidance was activated that drove the aircraft toward a circle of specified radius around the target. Different range commands were applied in order to get the aircraft to close in and move away from the target.

3. Results

The results from the hardware in the loop simulation showed that the guidance system worked extremely well, however the target tracking software had difficulty keeping track of the target for the entire length of the flight. The camera operator was forced to keep a close watch on the video feed and re-designate the target every time it was lost. Every time the target was lost, the range estimate became inaccurate, and it took a significant period of time to regain its accuracy. Despite the problems associated with the tracking software losing the target, everything else worked extremely well.

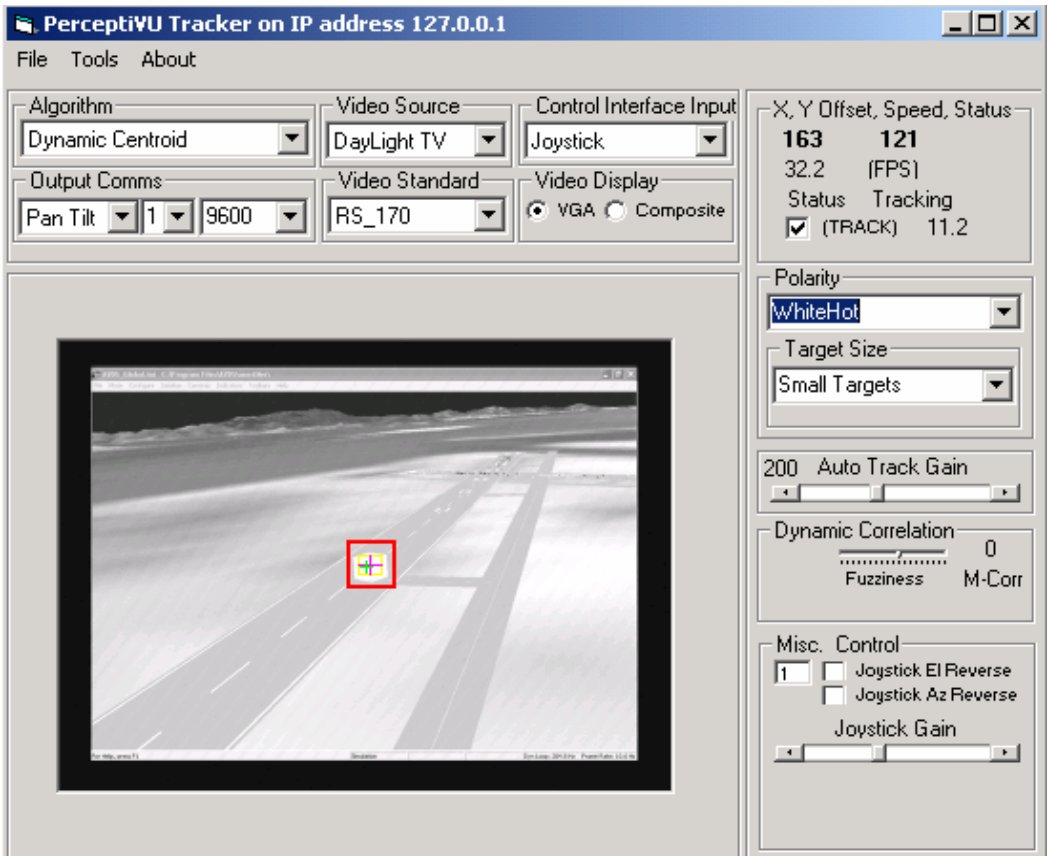

Figure 45. PerceptiVU Software Tracking a Target Created With AVDS Software

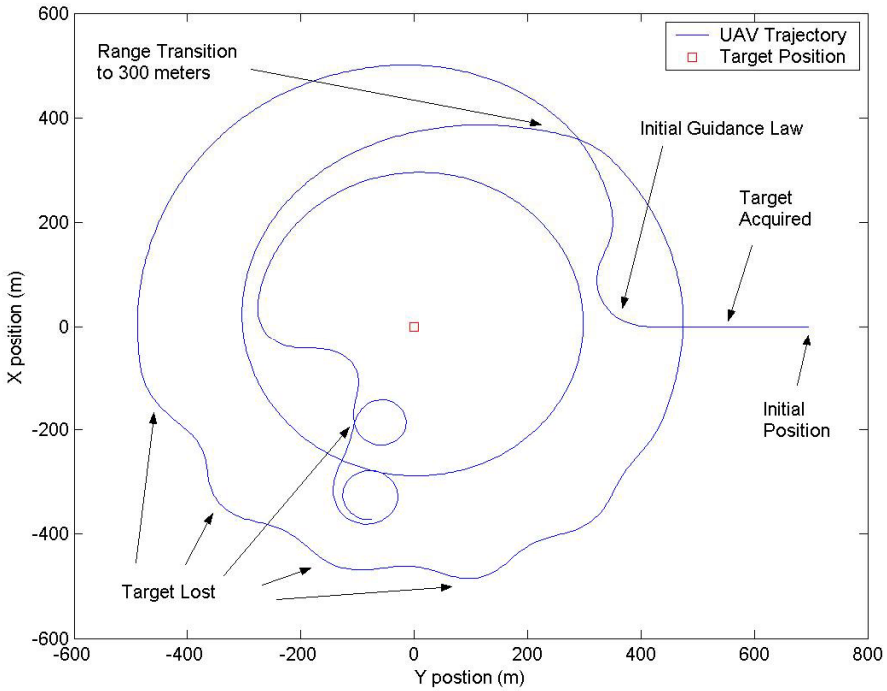

Figure 46. Horizontal Trajectory of UAV in Hardware in the Loop Simulation Showing Poor Performance Due to the Target Being Lost by the PerceptiVU Tracking Software

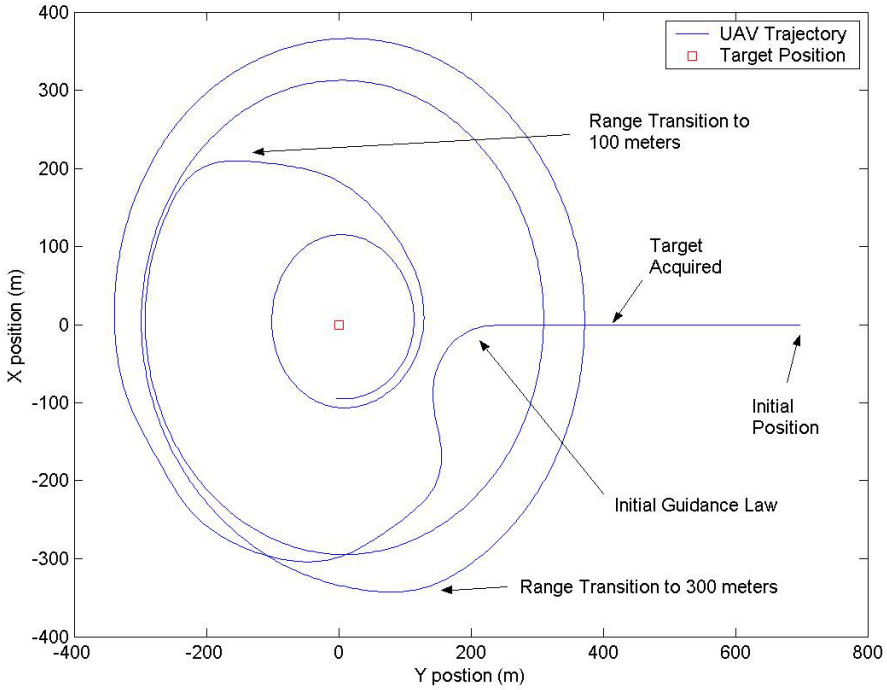

Figure 47. Horizontal Trajectory of UAV in Hardware in the Loop Simulation Showing Good Performance

4. Conclusions

The results of the hardware in the loop simulation allow for some conclusions to be made. The Kalman filters for range estimation need to be de-activated whenever the target is lost from the tracking software. This will prevent the estimates from being affected when inaccurate data is being obtained. This can likely be implemented with some sort of logic function that observes the signals from the PerceptiVU software. Also, the results of this simulation further emphasize the need for high quality video in order to make target tracking a possibility.

D. FLIGHT TESTING

Actual flight testing is the last step in verifying the guidance law and proving that autonomous visual tracking is possible with the equipment that is used. Flight tests are useful in verifying that everything works in a real-life situation, however they are not very effective in determining what the problems are in systems that are not working. This is the last step in the testing process, and all of the previous testing should have positive results before it is even attempted.

At this time the hardware has not been fully assembled so flight tests cannot be conducted. This will occur once all of the hardware systems have been developed.

1. Complete System for Flight Test

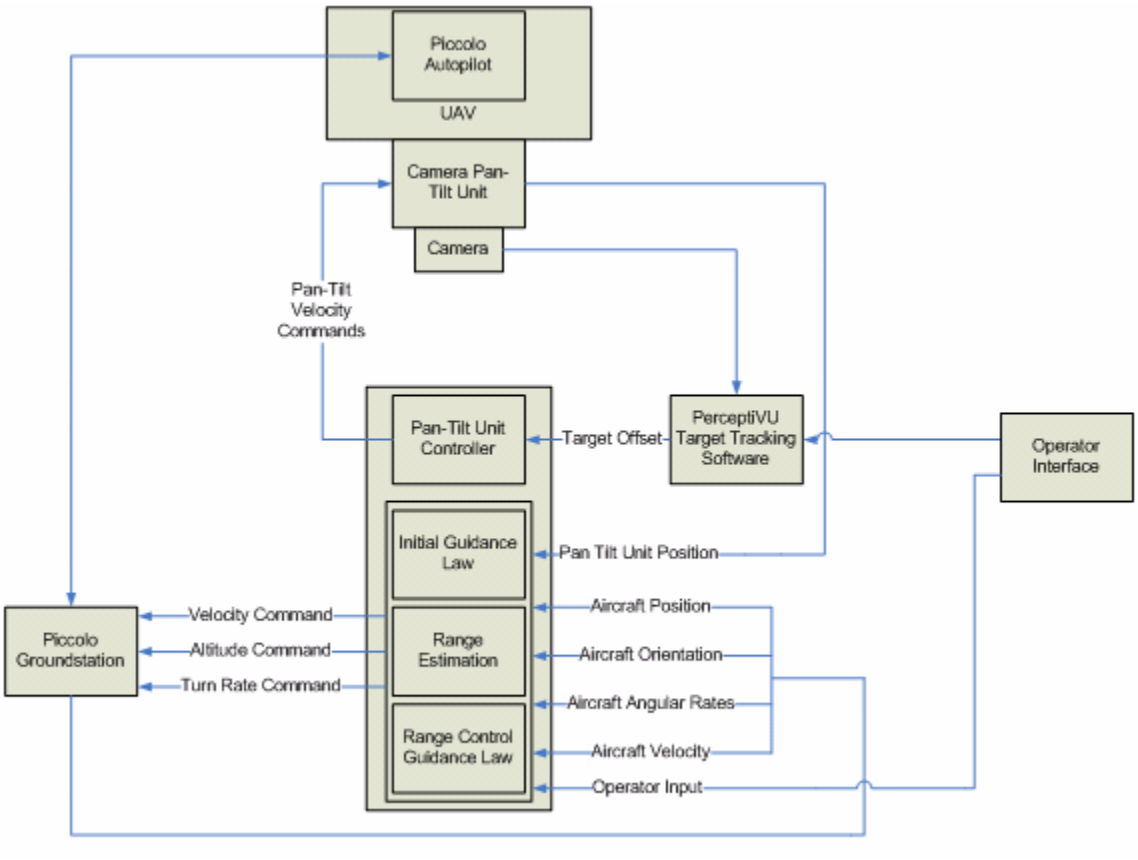

Figure 48. Setup for Actual Flight Test

VIII. CONCLUSIONS AND RECOMMENDATIONS

A. CONCLUSIONS

A control system for autonomous visual tracking of a stationary target was developed. This control system used the target offset from the PerceptiVU target tracking software to directly control the pan-tilt unit with proportional control. With this loop of the control system functioning, the aircraft trajectory could then be controlled to assist in tracking the target. The initial aircraft guidance control law that was developed drives the aircraft trajectory to a constant radius circle around the target. During this phase, the range to the target can be estimated using steady state Kalman filters, and once an estimate for range has converged, a range control guidance law can be activated. The range control guidance law drives the aircraft trajectory to a circle around the target with a radius equal to the desired range.

Initial tests conducted with Simulink models verify that the control laws developed work well, even in the presence of noise and disturbances. Closed loop testing with the PerceptiVU target tracking software also verified that the control laws work well. Currently the only limitation in the closed loop testing is the target tracking software's ability to keep a lock on the target.

B. RECOMMENDATIONS

1. Further Testing with Hardware in the Loop Simulations

Further testing should be done with hardware in the loop simulations, replacing the model of the pan-tilt unit with the actual device, and replacing the autopilot model with the actual Piccolo autopilot that will be used. With a comprehensive hardware in the loop simulation, almost all of the hardware will be used, and the only components that will remain untested are the actual aircraft and the camera.

2. Conduct Flight Tests

The only way to verify that the control system developed in this thesis will actually work for autonomous tracking of stationary targets is to conduct actual flight tests. This should be done as soon as all hardware has been developed and the results should be carefully examined to determine where the control laws could be improved.

3. Position Estimation of Target

In the current setup, the range to the target is the only estimate, although it would be possible to create an estimate of the target position. Although estimating the position of the target instead of range would require more computational work, it would create a more robust system overall. If the image tracking software loses the target in the current setup, the guidance law becomes immediately useless and a human operator would have to take over control of both the UAV and the camera. If the position of the target was estimated, the camera could be aimed at the estimated position when the tracking software loses the target. Therefore, the UAV could still follow the trajectory governed by the guidance law, and if the estimates were accurate, the target would still be held in the field of view of the camera. The operator could then re-designate the target and the system would continue to operate autonomously.

4. Develop Ability to Track Moving Targets

Upon successful testing of the current control system for stationary targets, it should be modified to have the ability to track moving targets as well. For slow moving targets the aircraft trajectory would still be driven to a circle around the target, however as target speed increased the trajectory should likely be driven to S-turns following the target, and finally to a straight line directly following the target.

LIST OF REFERENCES

1. Siouris, George M. *Aerospace Avionics Systems: A Modern Synthesis.* New York: Academic Press, 1993.
2. Etkin, Bernard and Lloyd Duff Reid. *Dynamics of Flight, Third Edition*. New York: John Wiley and Sons, 1995.
3. Ogata, Katsuhiko. *Modern Control Engineering, Fourth Edition*. New Jersey: Prentice Hall, 2002.
4. Franklin, Gene F., J. David Powell and Abbas Emani-Naeini. *Feedback Control of Dynamic Systems*. New Jersey: Prentice Hall, 2002.
5. Whang, Ick-Ho, Won-Sang Ra and Jo-Young Ahn. "A Modified Weighted Least Squares Range Estimator for ASM Application." Agency for Defense Development, Korea.
6. Vagilienti, Bill and Ross Hoag. *Piccolo System User Guide*. Cloud Cap Technologies, 2004.
7. Grewal, Mohinder S. and Angus P. Andrews. *Kalman Filtering: Theory and Practice Using MATLAB*. New York: John Wiley and Sons, 2001.
8. Lewis, Frank L. *Optimal Estimation with an Introduction to Stochastic Control Theory.* New York: John Wiley and Sons, 1986.
9. Schmidt, L., *Introduction to Aircraft Flight Dynamics*. American Institute of Aerodynamics and Astrodynamics, 1998.
10. Lizarraga, Mariano. *Autonomous Landing System for a UAV*. Master's Thesis, Naval Postgraduate School, Monterey, CA, 2004.

THIS PAGE INTENTIONALLY LEFT BLANK

INITIAL DISTRIBUTION LIST

1. Defense Technical Information Center
   Ft. Belvoir, Virginia

2. Dudley Knox Library
   Naval Postgraduate School
   Monterey, California

3. Dr. Isaac Kaminer
   Naval Postgraduate School
   Monterey, California

4. Dr. Ick Ho Whang
   Naval Postgraduate School
   Monterey, California

5. Dr. Vladimir Dobrokhodov
   Naval Postgraduate School
   Monterey, California

6. Dr. Anthony Healey
   Chairman, Department of Mechanical and Astronautical Engineering
   Naval Postgraduate School
   Monterey, California